

**Univerzita Pardubice**  
**Fakulta ekonomicko-správní**  
**Ústav systémového inženýrství a informatiky**

**Návrh a tvorba databáze v prostředí vybrané firmy**

Pavla Vaníčková

Bakalářská práce

2012

Univerzita Pardubice  
Fakulta ekonomicko-správní  
Akademický rok: 2011/2012

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Pavla Vaničková**  
Osobní číslo: **E09125**  
Studijní program: **B6209 Systémové inženýrství a informatika**  
Studijní obor: **Informatika ve veřejné správě**  
Název tématu: **Návrh a tvorba databáze v prostředí vybrané firmy**  
Zadávající katedra: **Ústav systémového inženýrství a informatiky**

### Z á s a d y p r o v y p r a c o v á n í :

- 1) Úvod, charakter ukládaných dat a popis tvorby těchto dat
- 2) Analýza současného stavu ukládání a správy dat ve firmě
- 3) Výběr a popis konkrétního databázového systému
- 4) Návrh konkrétní databáze podle potřeb firmy
- 5) Implementace vytvořeného modelu
- 6) Závěr, zhodnocení vytvořené databáze

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] ŠIMONOVÁ, Stanislava; PANUŠ, Jan. Databázové systémy I : pro kombinovanou formu studia. Pardubice : Univerzita Pardubice, 2007. 106 s. ISBN 978-80-7194-988-6.
- [2] POKORNÝ, Jaroslav ; HALAŠKA, Ivan. Databázové systémy. Praha : Vydavatelství ČVUT, 2003. 148 s. ISBN 80-01-02789-9.
- [3] POKORNÝ, Jaroslav. Databases in the 3rd Millennium: Trends and Research Directions. Journal of Systems Integration [online]. 2010, No 1-2, [cit. 2011-06-27]. Dostupný z WWW: <<http://www.si-journal.org/index.php/JSI/article/viewFile/25/36>>.
- [4] NOSQL Databases [online]. 2009 [cit. 2011-06-27]. Dostupné z WWW: <<http://nosql-database.org/>>.
- [5] ELOV Elektro [online]. 2010 [cit. 2011-06-27]. Dostupné z WWW: <<http://www.elov.cz/>>.


Vedoucí bakalářské práce:

  
Ing. Milan Tomeš

Ústav systémového inženýrství a informatiky

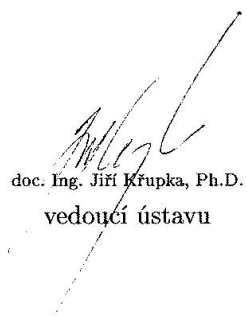
Datum zadání bakalářské práce: **3. října 2011**

Termín odevzdání bakalářské práce: **30. dubna 2012**

  
doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.

  
doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 3. října 2011

## **Prohlášení**

Prohlašuji, že jsem tuto práci vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako Školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 23. 4. 2012

Pavla Vaníčková

## **Poděkování:**

Tímto bych ráda poděkovala svému vedoucímu práce Ing. Milanu Tomešovi za jeho odbornou pomoc, cenné rady a poskytnuté materiály, které mi pomohly při zpracování bakalářské práce.

Dále bych ráda poděkovala panu Josefu Vaníčkovi za veškeré informace a čas, který mi věnoval při tvorbě této práce.

## **Anotace**

*Práce se zabývá návrhem a implementací databáze pro konkrétní firmu, která zatím nepoužívá žádný způsob elektronického ukládání dat. Vytvořená databáze by měla zpřehlednit ukládání informací o zakázkách a zefektivnit pozdější práci s těmito daty.*

## **Klíčová slova**

*Databáze, databázový model, konceptuální model, relační model dat, transformace, normalizace, entita, atribut, relace*

## **Title**

Design and production of database in surroundings of a selected company

## **Annotation**

*The work deals with the design and implementation of a database for a particular company which currently doesn't use any method of electronic storage. The created database should increase transparent storage of information of contracts and streamline future work with these data.*

## **Keywords**

*Database, database model, conceptual model, relation data model, transformation, normalization, entity, attribute, relation*

# Obsah

Úvod.....	10
<b>1. Teoretická východiska.....</b>	<b>11</b>
1.1 Databázové systémy .....	11
1.1.1 Relaçní databázový model .....	12
1.2 Konceptuální modelování.....	13
1.2.1 Entity a atributy .....	14
1.2.2 Klíçe .....	14
1.2.3 Vztahy mezi entitami.....	15
1.2.4 E-R diagram.....	16
1.2.5 MIN-MAX notace .....	17
1.3 Relaçní model dat.....	18
1.3.1 Atribut, n-tice, doména.....	18
1.3.2 Relace .....	19
1.3.3 Klíçe .....	19
1.3.4 Relaçní integrita.....	20
1.4 Transformace .....	21
1.4.1 Reprezentace vztahu 1:1.....	21
1.4.2 Reprezentace vztahu 1:N .....	22
1.4.3 Reprezentace vztahu M:N.....	22
1.5 Normalizace .....	22
1.5.1 První normální forma .....	22
1.5.2 Druhá normální forma .....	23
1.5.3 Třetí normální forma .....	23
<b>2. Analýza problematiky .....</b>	<b>24</b>
2.1 Základní informace o firmě .....	24
2.2 Analýza prostředí .....	25

2.3	Požadavky na databázi .....	26
<b>3.</b>	<b>Tvorba konceptuálního návrhu .....</b>	<b>27</b>
3.1	Vymezení entit .....	27
3.2	Přidělení atributů .....	28
3.3	Vztahy mezi entitami .....	30
<b>4.</b>	<b>Tvorba relačního modelu dat.....</b>	<b>33</b>
4.1	Návrh relací.....	33
4.2	Užití normálních forem .....	36
<b>5.</b>	<b>Implementace relačního modelu dat.....</b>	<b>40</b>
5.1	Výběr softwarového nástroje .....	40
5.1.1	<i>Microsoft Access 2007</i> .....	40
5.2	Tvorba tabulek .....	41
5.2.1	<i>Definování polí</i> .....	41
5.2.2	<i>Definování relací</i> .....	43
5.3	Vkládání dat .....	44
5.4	Formuláře, dotazy, sestavy .....	46
5.4.1	<i>Formuláře</i> .....	46
5.4.2	<i>Dotazy</i> .....	47
5.4.3	<i>Sestavy</i> .....	47
<b>6.</b>	<b>Zhodnocení .....</b>	<b>49</b>
	<b>Závěr .....</b>	<b>50</b>
	<b>Použitá literatura .....</b>	<b>51</b>



## Seznam obrázků

Obrázek 1: Terminologie databázových systémů.....	12
Obrázek 2: Grafické vyjádření E-R diagramu.....	17
Obrázek 3: Grafické vyjádření MIN – MAX notace.....	17
Obrázek 4: E-R diagram modelované reálie.....	32
Obrázek 5: Vztah Nabídka – Odběratel.....	33
Obrázek 6: Vztah Zakázka – Odběratel.....	33
Obrázek 7: Vztah Zakázka – Smlouva.....	34
Obrázek 8: Vztah Zakázka – Faktura.....	34
Obrázek 9: Vztah Zakázka – Pracovník.....	34
Obrázek 10: Vztah Zakázka – Předávací protokol.....	35
Obrázek 11: Vztah Zakázka – Objednávka.....	35
Obrázek 12: Vztah Dodavatel – Objednávka.....	35
Obrázek 13: Vytvoření nové tabulky.....	41
Obrázek 14: Návrhové zobrazení a datové typy.....	42
Obrázek 15: Relace mezi tabulkami.....	44
Obrázek 16: Vytvoření nového formuláře.....	46
Obrázek 17: Vytvoření nové sestavy.....	48

## Seznam tabulek

Tabulka 1: Analogie pojmů.....	18
Tabulka 2: Entity a jejich atributy.....	28
Tabulka 3: Entity, atributy a jejich identifikátory.....	30
Tabulka 4: Přehled relací a názvů atributů.....	36
Tabulka 5: Relace upravené podle 1NF.....	37
Tabulka 6: Relace upravené podle 3NF.....	39

## Seznam příloh

Příloha A – Formulář pro vkládání dat o zakázkách

Příloha B – Ukázky návrhového zobrazení dotazů

Příloha C – Ukázky sestav

## Seznam zkratk

1NF	První normální forma
2NF	Druhá normální forma
3NF	Třetí normální forma
CK	Cizí klíč
DB	Databáze
DBS	Databázový systém
DIČ	Daňové identifikační číslo
E-R diagram	Diagram entit a vztahů (Entity Relationship Diagram)
IČO	Identifikační číslo organizace
MS Office	Microsoft Office
OLE	Object Linking and Embedding
PSČ	Poštovní směrovací číslo
SQL	Strukturovaný dotazovací jazyk (Structured Query Language)
SŘBD	Systém řízení báze dat

## Úvod

I přes dnešní rozmach informačních technologií a moderních způsobů ukládání a zpracovávání dat se stále najdou firmy, které neumějí tyto technologie správně používat a nejsou schopné si s jejich pomocí ulehčovat a zefektivňovat práci. Přitom kvalitní správa dat může přispět k celkovému zlepšení firemní situace. Ve chvíli, kdy jsou data správně ukládána a spravována, může firma plně využívat všech dostupných informací, které za delší dobu nashromáždila. Pomocí softwarového nástroje mohou být nad daty prováděny různé operace, data mohou být seskupována, filtrována, upravována pro tiskové výstupy atd. Při správném používání všech možností funkční databáze může firma získat ucelený a přehledný pohled na současný stav podniku a případně na základě zjištěných informací podniknout příslušné změny do budoucna.

Cílem této práce je vytvořit funkční databázi pro konkrétní firmu, která v současnosti nemá zavedený žádný způsob elektronického ukládání dat. Vytvořená databáze by měla přispět k lepší organizaci dat a usnadnění získávání potřebných informací.

Proces tvorby databáze začíná modelováním firemní reálie a je dále upravován podle neměnných pravidel. První část práce se zabývá teorií databází a popisuje postup při tvorbě návrhu databáze. Následuje samotná tvorba návrhu databáze pro vybranou firmu, podle teoretických východisek. Vytvořený model je následně implementován v softwarovém nástroji a konzistentnost vytvořené databáze je ověřena na konkrétních údajích firmy. Uvedena jsou i doporučení pro budoucí ukládání dat a jejich správu tak, aby výsledná databáze byla pro firmu co nejefektivnější.

# 1. Teoretická východiska

V této kapitole budou stručně popsány teoretické poznatky týkající se databází a jejich tvorby. Pozornost byla zaměřena pouze na tu část problematiky, která je nezbytná pro praktické využití v práci.

## 1.1 Databázové systémy

Databázi (DB) nebo jinak řečeno bázi dat můžeme definovat jako navzájem vázaná data, která jsou uložena v nějaké paměti tak, že nejsou závislá na žádném programu, který by je měl používat. Zároveň platí, že přidávání nových dat, úprava nebo výběr dat existujících v rámci jedné databáze jsou řízeny centrálně.[12]

Systém řízení bází dat (SŘBD) je speciální programové vybavení, jehož prostřednictvím je organizována centrální správa databáze. Společně s nějakou databází potom tvoří databázový systém (DBS). Tento vztah můžeme jednoduše vyjádřit sloganem

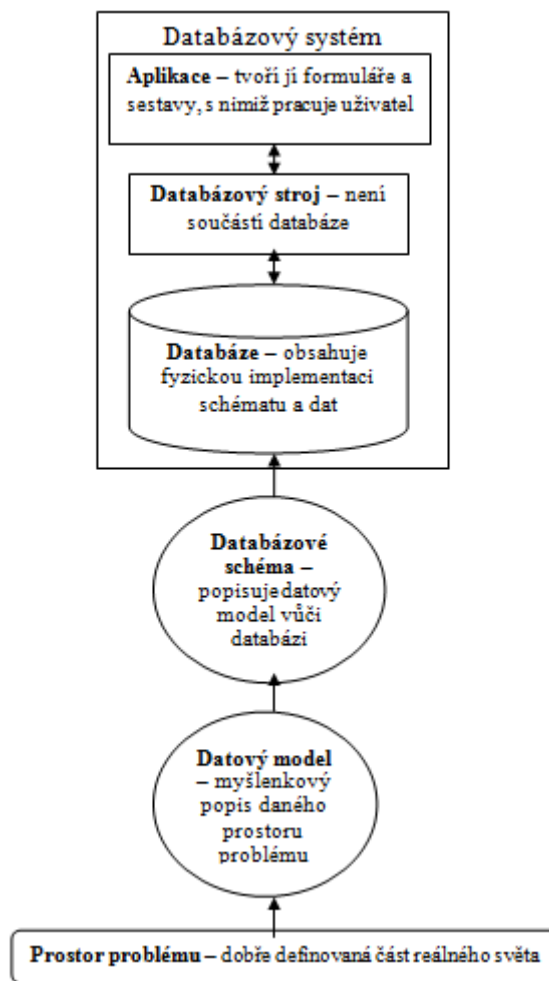
$$DB + SŘDB = DBS [11]$$

Databázový model je charakterizován jako soubor pojmů používaných k modelování, jinak řečeno databázový model může být například matematický nebo formální aparát. Výstupem databázového modelování je jistý popis struktury dat, který se označuje jako databázové schéma.[15]

Databázový model může být popsán také jako nástroj umožňující reprezentovat strukturu a funkcionalitu databáze. Definuje schéma databáze, které udává organizaci dat, způsob ochrany a zajištění správnosti dat a operace s daty. Podle typu modelovaných vztahů mezi jednotlivými záznamy se rozlišují hierarchický, síťový, relační a objektově orientovaný model. Databázové schéma může být tvořeno pomocí strukturované nebo objektově orientované metody.[2]

Databázové stroje jsou mechanismy, které provádějí fyzickou manipulaci s daty, tzn., ukládají je na disk a opětovně načítají po jejich vyžádání. Dvěma nejznámějšími databázovými stroji jsou Microsoft Jet a SQL Server. Tyto dva typy se zásadně liší svojí architekturou, Microsoft Jet je typem „stolního“ stroje, který se hodí pro malé a střední systémy. SQL Server pracuje na architektuře klient/server a slouží pro středně velké až rozsáhlé systémy.[13]

Na obrázku 1 je zobrazen databázový systém, a jak na sebe navazují jednotlivé výše popsané prvky.



**Obrázek 1:** Terminologie databázových systémů.

*Zdroj: upraveno podle [13]*

### 1.1.1 Relační databázový model

Tato podkapitola je zaměřena na stručné popsání relačního databázového modelu, který je v současné době nejpoužívanějším modelem. Relační model bude také použit u databáze, jejíž vytvoření je cílem této práce. Informace v této kapitole byly čerpány ze zdroje [4].

Relační databázi v roce 1969 představil Dr. Edgar F. Codd. Jako matematik se pokusil existující problémy s redundací dat, nízkou integritou a přílišnou závislostí databázových struktur na jejich fyzické realizaci vyřešit pomocí matematických technik. Model založil na teorii množin a predikátové logice prvního řádu. Jméno modelu je odvozeno z pojmu „relace“, který je součástí teorie množin.

V relačních databázích jsou data uložena v relacích, které konečný uživatel vidí jako tabulky. Každá taková relace se skládá z uspořádaných n-tic a atributů. Skutečné uspořádání n-tic nebo atributů v databázi je nepodstatné a každá n-tice je v tabulce identifikována atributem obsahujícím unikátní hodnotu.

Data z relačních databází se získávají pomocí jazyka SQL (strukturovaný dotazovací jazyk), jde o standardní jazyk využívaný pro vytváření, modifikaci a udržování databáze a vytváření dotazů. Požadovaná data jsou získávána pomocí SQL dotazů, které může buď uživatel zadat přímo, nebo pomocí grafického nástroje. K práci s databází není vždy nutné znát jazyk SQL, ale je dobré znát alespoň základy, což uživateli usnadní pochopení a odstraňování problémů při vytváření dotazů pomocí grafických nástrojů.

Základními částmi SQL dotazu tedy jsou: SELECT...FROM, klauzule WHERE a klauzule ORDER BY. SELECT se používá pro specifikaci atributů, které mají být užity v dotazu, klauzule FROM určuje tabulky, ve kterých se tyto atributy nacházejí. Klauzule WHERE slouží pro filtrování vrácených n-tic podle jednoho nebo více atributů. Vrácené n-tice lze také seřadit vzestupně či sestupně pomocí klauzule ORDER BY.

Mezi hlavní výhody relační databáze patří zabudovaná víceúrovňová integrita, logická a fyzická nezávislost dat na databázové aplikaci, garantovaná konzistence a přesnost dat daná různými úrovněmi integrity a snadné získávání dat.

## **1.2 Konceptuální modelování**

Konceptuální modely umožňují vytvoření popisu dat v databázi, tzn. nějakého konceptuálního schématu, které bude nezávislé na fyzickém uložení databáze. Tento popis by měl zároveň co nejlépe zachycovat konceptuální pohled člověka na daný výsek reálného světa.[11]

Jinak řečeno jde o model umožňující popsat a zobrazit objekty a jejich vzájemné vztahy v databázi z hlediska jejich chování a významu. Výsledkem je pak schéma, které je obecně aplikovatelné v jakémkoli technicko-programovém prostředí.[6]

Dále se práce bude zabývat nejznámějším konceptuálním modelem, E-R modelem, jedná se o model entit a jejich vztahů a je určený pro účely počátečního návrhu databáze.[12]

### 1.2.1 Entity a atributy

Účelem databází je ukládat informace o určitých objektech. V databázovém názvosloví se tyto objekty označují jako entity.[12]

„Entita je rozlišitelný a identifikovatelný objekt reality. Tedy je to objekt reálného světa, který je schopen nezávislé existence a je jednoznačně odlišitelný od ostatních objektů.“[15]

Každý výskyt entity musí být jednoznačně identifikovatelný, to znamená, že každá entitní množina musí mít uveden identifikátor, který zajistí jednoznačnou identifikaci entit v množině.[15]

Na základě podobnosti entit se provádí jejich sdružování do entitních množin, tzv. typů entit. Výskyt entity je potom výskyt objektu entitního typu.[15]

„Atributem budeme rozumět funkci přiřazující entitám či vztahům hodnotu (zde popisného typu), určující některou podstatnou vlastnost entity nebo vztahu.“ [11]

Atribut je tedy vlastnost entity, která jí blíže charakterizuje. Dle [15] můžeme rozlišovat atributy:

- podle volitelnosti – atribut má hodnotu pro každý výskyt entity (totální atribut) nebo nemusí mít hodnotu pro každý výskyt entity (parciální atribut)
- podle odvoditelnosti – atribut nelze odvodit z jiných atributů (základní atribut) nebo lze atribut získat odvozením z jiných atributů (odvoditelný atribut)

### 1.2.2 Klíče

Klíč je minimální množina atributů, zajišťující jednoznačnou identifikaci výskytů entity, je tedy identifikátorem entity. Často je nazýván identifikační klíč. Tento identifikátor by dle [15] měl být:

- jednoznačný – jedna hodnota identifikátoru určuje právě jeden výskyt entity
- úplný – pro každý výskyt entity existuje právě jedna hodnota identifikátoru
- minimální – v případě, že je identifikátor složený z více atributů, pak nesmí obsahovat žádnou podmnožinu, která je jednoznačná a úplná
- stabilní – během života entity nemění svojí hodnotu

Druhy klíčů, které jsou dle [15] rozlišovány:

- kandidátní klíč – klíč, který je kandidátem na identifikační klíč, nemusí se jím ale stát,

- cizí klíč – atribut, který je v jiné entitě identifikačním klíčem nebo jeho částí,
- alternativní klíč – minimální množina atributů zajišťující jednoznačnou identifikaci výskytů entity, která nebyla zvolena za identifikační klíč,
- sekundární klíč – neprázdná množina atributů, které jsou důležité pro přístup k datům reprezentovaným entitou.

### 1.2.3 Vztahy mezi entitami

Jednotlivé entity mají mezi sebou konkrétní vztahy. Vztah tedy vyjadřuje reálnou vazbu mezi dvěma či více entitami a vyjadřuje informaci, kterou nelze odvodit z atributů jednotlivých entit. Vztah má jméno, které vyjadřuje jeho podstatu z hlediska obou entit i z hlediska vztahu jako takového.[15]

Aby byla zajištěna správnost dat v databázi, je třeba definovat integritní omezení. Integritní omezení je tvrzení, které říká, co má platit o objektech v dané realii. Jestliže jsou integritní omezení špatně definována, může dojít ke ztrátě integrity databázového systému.

Integritní omezení vztahů dle [15]:

- kardinalita vztahu – maximální a minimální počet výskytů entity v určitém vztahu,
- volitelnost (parcialita) vztahu – povinné či nepovinné členství entity ve vztahu
- výlučnost (exkluzivita) vztahu – zda pro jeden výskyt entity může být realizován právě jeden z vztahů vzájemně výlučných
- externí identifikace (slabý vztah) – vyjadřuje externí identifikační závislosti určité entity v určitém vztahu na jiné (k plné identifikaci entity nestačí její vlastní atributy)

#### **Kardinalita vztahů**

Kardinalitu vztahu můžeme vyjádřit tvrzením, že entita jednoho typu jednoznačně určuje (neurčuje) entitu druhého typu. Případně, že entita jednoho typu je (není) determinantem entity druhého typu.[5]

Kardinalita znamená maximální a minimální počet výskytů entity v daném vztahu. Tyto vztahy lze dále klasifikovat dle [14]:

- vztah 1:1 – každá entita jedné množiny je spojena vztahem s nejvýše jednou entitou druhé množiny



- vztah 1:N – mezi množinami E1 a E2 reprezentuje situaci, kdy každá entita z E1 je spojena vztahem s žádnou či více entitami z E2, ale každá entita z E2 je spojena vztahem s nejvýše jednou entitou z E1
- vztah M:N – je nejobecnější, nejsou kladena žádná omezení na množinu dvojic entit spojených příslušným vztahem.

### **Členství ve vztahu**

„Některá organizační pravidla modelované reality určují, že každý výskyt entity musí být do vztahu zapojen, jindy je dovoleno, aby některé výskyty elitního typu existovaly mimo vztah.“[11]

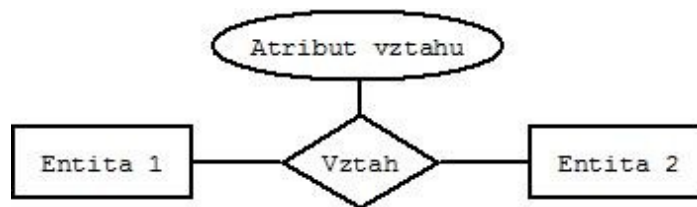
Povinné (totální) členství ve vztahu, je když vztah musí vzniknout, nepovinné (parciální), kdy vztah vzniknout může, ale nemusí. Rozhodnutí o tom, zda vznikne vztah nebo ne musí být výsledkem zkoumání dané reality, není možné si tuto skutečnost vymyslet.

#### **1.2.4 E-R diagram**

Třídy entit, jejich atributy a vztahy mezi nimi je možné v databázovém modelu ilustrovat pomocí diagramu entit a vztahů, který se jinak nazývá E-R diagram (Entity-Relationship diagram).[12]

Nejedná se však o jeden přesně daný diagram, jedná se spíše o rodinu E-R diagramů, ve které je celá možnost variabilit zobrazení. Nejdůležitější je na začátku si stanovit jeden přesně daný typ diagramu a ten používat po celou dobu.[15]

V práci bude použito grafické zobrazení, kde entitní typy jsou reprezentovány obdélníky, vztahové typy pomocí kosočtverců a čarami je vyjádřeno, které entitní typy jsou zapojeny do jednotlivých typů vztahů. Pro zjednodušení a větší přehlednost bude použit diagram, který nezobrazuje jednotlivé atributy entit. Grafické zobrazení atributu jako oválu bude použito pouze v případě, že daný atribut bude charakterizovat vztah. Použitá grafika E-R diagramu je zobrazena na obrázku 2.



**Obrázek 2:** Grafické vyjádření E-R diagramu

*Zdroj: vlastní zpracování*

### 1.2.5 MIN-MAX notace

Dle [15] [15] je jedním z možných způsobů grafického vyjádření integritních omezení v rámci E-R diagramu tzv. Min-Max notace. Hodnota Min vyjadřuje minimální počet výskytů entity a nábývá hodnot:

- Min = 0 – vyjadřuje nepovinné členství ve vztahu, vyjadřuje tedy možnost vzniku vztahu,
- Min = 1 – vyjadřuje povinné členství ve vztahu, vyjadřuje tedy nutnost vzniku vztahu.

Hodnota Max vyjadřuje maximální počet výskytů entity a nábývá hodnot:

- Max = 1 – znamená nejvýše jeden výskyt
- Max = N – znamená, že výskytů je více

Na obrázku 3 je zobrazeno grafické vyjádření MIN-MAX notace, které bude dále používáno v této práci. Uvedený příklad říká, že Entita1 má nepovinné členství ve vztahu s maximálně jedním výskytem a Entita2 má povinné členství ve vztahu s jedním nebo více výskyty.



**Obrázek 3:** Grafické vyjádření MIN – MAX notace

*Zdroj: vlastní zpracování*

## 1.3 Relační model dat

Popis a výhody relačního modelu dat jsou již uvedeny v podkapitole 1.1.1, proto se tato kapitola zaměří na vysvětlení základních pojmů užívaných v tomto modelu. Jedná se především o pojmy relace, n-tice, atribut, doména, klíče a relační integrita. V tabulce 1 je zobrazena analogie základních pojmů.

**Tabulka 1:** Analogie pojmů

Relační pojem	Reprezentace	Souborová analogie
relace	tabulka	soubor
n-tice	řádek	záznam
atribut	sloupec	položka

*Zdroj: upraveno podle [7]*

### 1.3.1 Atribut, n-tice, doména

Atribut může být definován jako sloupec relace označený unikátním názvem a doménou. Samotná relace je pak určena svojí množinou atributů, které jsou v rámci této relace definovány unikátními jmény, která zabezpečí správné adresování při přístupu k hodnotám reprezentovaným v samotné relaci.[7]

Pojem atribut byl již zmíněn v kapitole 1.2.1, avšak definice uvedená ve zmíněné kapitole je zcela rozdílná a je nezbytné tyto dva pojmy od sebe odlišovat. Atribut relace blíže popisuje hodnoty ukládané do příslušné relace a je určený i datovým typem nebo dalšími integritními omezeními týkajícími se hodnot, kterých atribut může nabývat.[7]

N-tice představuje unikátní výskyt entity dané relace. Je složena z celé řady atributů v relaci, bez ohledu na to, zda tyto atributy obsahují nějaké hodnoty nebo ne. Každá n-tice je v databázi identifikována unikátní hodnotou v primárním klíči dané n-tice.[4]

Doména je definovaná jako pojmenovaná množina skalárních hodnot stejného typu. Skalární hodnota je pak taková hodnota, která reprezentuje nejmenší sémantickou jednotku dat. Z hlediska implementace může být doména definovaná jako interval nebo množina. Domény mají velký význam při relačních operacích.[7]

### 1.3.2 Relace

Dle [7] se relace  $R$  nad množinou domén  $D_1, D_2, \dots, D_n$  skládá ze dvou částí, hlavičky a těla:

- hlavička – je stálá množina atributů  $A_1, A_2, \dots, A_n$ , o které platí, že každému atributu  $A_i$  odpovídá právě jedna z domén  $D_i$  kde  $i=1,2, \dots, n$ , toto se dá také označit  $(A_i:D_i)$
- tělo – množina  $n$ -tic, která se mění v čase a kde každou  $n$ -tici tvoří množina dvojic  $(A_i: v_{ij})$ , kde  $i=1,2, \dots, n$  a  $j=1,2, \dots, m$ , dále platí, že hodnota  $v_{ij}$  je jednou z hodnot dané domény  $D_i$  příslušející atributu  $A_i$ .

Dále dle [7] by každá relace měla splňovat tyto vlastnosti:

- neobsahuje duplicitní  $n$ -tice – tělo relace je matematická množina  $n$ -tic a podle definice neobsahuje duplicitní hodnoty, předpokladem splnění této vlastnosti je existence primárního klíče v relaci (viz 1.3.3)
- $n$ -tice jsou neuspořádané – ve smyslu shora dolů, což umožňuje zjednodušení a zrychlení prací nad relací
- $m$ -tice atributů jsou neuspořádané – ve smyslu zleva doprava, z toho vyplývá, že k jednotlivým sloupcům se přistupuje na základě identifikátoru (název atributu) ne podle jeho pozice
- hodnoty atributů jsou atomické – každý atribut má přidělenou vždy pouze jednu hodnotu, čímž se zabrání výskytu opakující se skupiny atributů. Jestliže je splněna tato vlastnost, relace je normalizovaná

### 1.3.3 Klíče

Teorie klíčů v relačním modelu dat je obdobná jako teorie klíčů entit při konceptuálním modelování (viz 1.2.2). Klíč je speciální atribut, který identifikuje danou  $n$ -tici. Nejvýznamnějšími klíči pak jsou primární klíč a cizí klíč.

Primární klíč je atribut, nebo skupina atributů, která jednoznačně identifikuje každou  $n$ -tici v relaci. Primární klíč může být složen ze dvou, nebo více atributů, pak je nazýván složeným primárním klíčem. Tento klíč je nejdůležitějším atributem relace a platí o něm:

- hodnota primárního klíče identifikuje danou  $n$ -tici v celé databázi
- atribut primárního klíče identifikuje danou relaci v rámci celé databáze

- primární klíč vynucuje integritu na úrovni relace a pomáhá zřizovat vztahy s ostatními relacemi v databázi [4]

Primární klíč je vybírán z množiny kandidátních klíčů, které musejí splňovat podmínky unikátnosti a minimálnosti. Zbylé kandidátní klíče, které nebyly vybrány za primární, se označují jako alternativní klíče. Atributy které nespádají ani do jedné z těchto skupin se nazývají neklíče.

Cizí klíč je množina atributů, které jsou v jiné relaci definované jako kandidátní nebo primární klíč. Pro cizí klíč by dle [7] mělo platit:

- cizí klíč může být množinou, tedy složený z více atributů
- každá hodnota cizího klíče, která se objeví v relaci, se musí vyskytovat jako hodnota primárního klíče v jiné relaci
- každý atribut tvořící cizí klíč musí mít definovanou stejnou doménu jako příslušný atribut kandidátního klíče
- cizí klíč je odkazem na řádek relace, kde hodnota kandidátního klíče je totožná s hodnotou cizího klíče
- cizí klíč je vyjádřením vztahu mezi dvojicí n-tic a reprezentuje spojení mezi relacemi

#### 1.3.4 Relační integrita

Integrita zajišťuje správnost dat v databázi. Správným užitím integrity při tvorbě modelu lze předejít chybám v ukládaných datech. Důsledkem chybně uložených dat může být vyhledávání informací nad daty, která jsou nepřesná nebo neplatná.

Dle [1] rozlišujeme tři druhy relační integrity:

- entitní integrita – integritní pravidlo, které se vztahuje na primární klíče relací a říká, že v žádné relaci nesmí mít atribut primární klíč prázdnou hodnotu
- referenční integrita – pravidlo vztahující se na cizí klíče, pokud existuje v relaci cizí klíč, musí buď hodnota cizího klíče odpovídat hodnotě některého záznamu v domovské tabulce, nebo musí mít cizí klíč prázdnou hodnotu
- integritní omezení – pravidla, která definují nebo omezují některé vlastnosti dat užívaných organizací

## 1.4 Transformace

Tato kapitola se zabývá transformací E-R modelu do relačního modelu dat. Teoretické informace použité v celé této kapitole jsou ze zdroje [11].

Jeden z možných způsobů práce s konceptuálním modelem je uspořádat atributy, příslušné jednomu elitnímu typu do schématu relace, kde primární klíč bude tvořen atributy, které odpovídají identifikačnímu klíči elitního typu a popisným typům atributů se přiřadí domény relace. Tímto způsobem se dají transformovat jednotlivé entity E-R modelu na počáteční relace.

Dalším krokem nezbytným pro transformaci je reprezentace vztahů, které byly identifikovány mezi entitami při modelování reality. Tyto vztahy je nutné reprezentovat s ohledem na danou kardinalitu a parcialitu. Transformace jednotlivých typů vztahů (viz 1.2.3) je popsána v následujících podkapitolách.

### 1.4.1 Reprezentace vztahu 1:1

Tento vztah vyjadřuje podle kardinality, že každá entita jedné množiny je spojena vztahem s nejvýše jednou entitou druhé množiny. Reprezentace vztahu je dále závislá na parcialitě, která udává povinnost (nepovinnost) členství entity v daném vztahu.

Mají-li oba členové vztahu povinné členství, vytvoří se pouze jedno schéma relace, které vznikne spojením původních entit. Oba původní klíče jsou unikátní, kterýkoliv z nich tedy může být primárním klíčem. Do této jedné relace jsou přidány i případné atributy vztahu.

Má-li jeden člen vztahu nepovinné členství a druhý povinné definujeme dvě schémata relací. Existenční závislost druhého typu na prvním vyjádříme tak, že k jeho schématu přidáme jako cizí klíč atributy odpovídající identifikačnímu klíči nezávislé entity. Případné atributy vztahu budou také přidány ke schématu závislé entity.

Mají-li oba entitní typy nepovinné členství ve vztahu, definují se tři schémata. Pro každý entitní typ a jedno schéma pro typ vztahu. Třetí schéma bude obsahovat klíče obou předchozích jako cizí klíče. Klíčem schématu může být libovolný z těchto dvou klíčů. Další atributy tohoto schématu budou odpovídat atributům vztahu.

### **1.4.2 Repräsentace vztahu 1:N**

Při transformaci vztahu 1:N je podstatná pouze parcialita entitního typu, který je ve vztahu determinantem, tzn. má pouze jeden výskyt v daném vztahu.

Má-li determinant vztahu povinnou účast, definují se dvě schémata relací, pro každý entitní typ jedno. Ke schématu determinantu je pak přidán jako cizí klíč identifikátor druhého entitního typu, který by neměl mít prázdnou hodnotu. Primárním klíčem schématu zůstává primární klíč determinantu.

Má-li determinant vztahu nepovinnou účast, definují se tři schémata. Pro každý entitní typ jedno a třetí schéma vztahové, které bude obsahovat jako cizí klíče identifikátory obou entitních typů a případné atributy daného vztahu. Primárním klíčem se stane cizí klíč, který odkazuje na identifikátor determinantu vztahu.

### **1.4.3 Repräsentace vztahu M:N**

Při reprezentaci tohoto typu vztahu již není parcialita podstatná. Vždy bez ohledu na členství ve vztahu vzniknou tři schémata relací. Pro každý entitní typ jedno a třetí pro vztah. Toto schéma pak bude obsahovat cizí klíče odkazující na identifikátory účastníků vztahu a případně další vztahové atributy. Primárním klíčem bude složený klíč z obou cizích klíčů.

## **1.5 Normalizace**

Jedná se o techniku používanou pro vytvoření sady relací s minimální redundací a která podporuje datové požadavky organizace. Normalizace se provádí jako sada testů na relaci, aby se určilo, zda jsou dodržena nebo porušena pravidla pro určitou normální formu. Existuje několik normálních forem, ale nejužívanější je první normální forma, druhá normální forma a třetí normální forma.[1]

### **1.5.1 První normální forma**

Relace je v první normální formě, jestliže každý její atribut obsahuje pouze atomické (dále nedělitelné) hodnoty. V případě, že některý atribut tuto podmínku nesplňuje, je nutné ho oddělit od původní relace a vytvořit novou relaci, která bude obsahovat hodnoty původního atributu a jako cizí klíč budou přidány hodnoty primárního klíče původní relace, aby pomocí nich mohlo vzniknout propojení těchto dvou relací.[17]

### **1.5.2 Druhá normální forma**

Relace se nachází ve druhé normální formě, jestliže je v první normální formě a každý neklíčový atribut je plně funkčně závislý na primárním klíči, a to na celém klíči, ne jen na některé jeho části. Z toho vyplývá, že tato normální forma se používá pouze v případě, že primární klíč je složený ze dvou nebo více atributů. Řešením je opět rozklad na dvě relace, kde v nově vzniklé relaci bude jako primární klíč ten atribut z původního primárního klíče, na kterém byl testovaný atribut závislý spolu s tímto atributem.[17]

### **1.5.3 Třetí normální forma**

Relace je ve třetí normální formě, splňuje-li předchozí dvě formy a žádný z jejich atributů není tranzitivně závislý na klíči. Tranzitivní závislost je taková závislost, mezi minimálně dvěma atributy a klíčem, kde jeden atribut je funkčně závislý na klíči a druhý je funkčně závislý na prvním. Řešením je opět rozklad na více relací. V nové relaci pak budou oba atributy, primárním klíčem se stane ten, který byl v původní relaci závislý pouze na klíči a ten také zůstane i v původní relaci jako cizí klíč, který umožní propojení těchto dvou relací.[17]



## 2. Analýza problematiky

V následující kapitole bude provedena analýza prostředí firmy Josef Vaniček – ELOV a na základě zjištěných nedostatků budou stanoveny požadavky na návrh databáze. Všechny informace uvedené v této kapitole byly získány jednak z internetových stránek firmy [3] a hlavně při osobním rozhovoru s majitelem firmy panem Josefem Vaničkem.

### 2.1 Základní informace o firmě

Firma Josef Vaniček – ELOV byla založena v roce 1992 původně jako firma zabývající se montáží vybraných elektrických zařízení. V současné době firma provádí dodávku a montáž sádrokartonů, sádrokartonových systémů, tepelné izolace, minerálních podhledů, cementových desek a střešních oken Velux. Na zateplování objektů získala firma certifikát odborného dodavatele v dotačním programu Zelená úsporám a provádí montáže zateplovacích systémů firmy LINZMEIER. Dále se firma stále zabývá elektroinstalacemi, a to převážně dodávkou a montáží silnoproudých a slaboproudých rozvodů, hromosvodných systémů, elektrických topných systémů Fenix. Firma také provádí revize elektrických zařízení, strojů a ručního přenosného nářadí. Jako doplňkovou činnost provozuje firma prodej veškerých světelných zdrojů, průmyslových svítidel a svítidel pro veřejné osvětlení.

Firma sídlí na adrese Vojnův Městec 301 Žďár nad Sázavou. I přes širokou nabídku poskytovaných služeb firma nevlastní žádné skladovací prostory. Materiál je vždy objednáván až podle potřeby konkrétní zakázky přímo na místo realizace.

Majitelem firmy je pan Josef Vaniček. V současné době má firma dva stálé zaměstnance. Pana Milana Procházku, který se stará o chod doplňkového prodeje světelných zdrojů a jednoho pomocného dělníka, který přímo pracuje na zakázkách. Při realizaci zakázky jsou pak nájímáni další pracovníci, většinou fyzické osoby pracující na živnostenský list, a to vždy podle konkrétního typu prováděných prací.

Účetnictví provádí externí účetní najatá firmou, vede však jen evidenci přijatých a vydaných dokladů. Fakturaci a evidenci provedených zakázek si spravuje majitel firmy sám.

## 2.2 Analýza prostředí

Vedení evidence zakázek je pro firmu důležité ze dvou hledisek:

- obsahuje údaje, které slouží jako podklad pro vyplňování dokumentů pro finanční úřad
- obsahuje údaje, které poskytují majiteli informace o vykonané práci a obratu firmy

I přes zjevnou důležitost ukládaných informací nevlastní firma žádný softwarový nástroj na ukládání těchto dat. Evidence je vedena papírovou formou, kdy majitel zakládá jednotlivé potřebné dokumenty týkající se dané zakázky. Tyto dokumenty jsou vytvářeny především majitelem firmy pomocí aplikací z balíku MS Office. Mezi tyto dokumenty patří:

- nabídka – jedná se o dokument, který majitel firmy zasílá odběrateli na základě jeho poptávky a obsahuje kompletní návrh prací, tak jak je firma schopná zakázku uskutečnit i s celkovým finančním přehledem, tento dokument zpracovává majitel převážně v MS Excel
- smlouva – v případě, že odběratel přijme nabídku firmy, vzniká smlouva o dílo, její přesná podoba je závislá na konkrétním typu prováděných prací, na tvorbu smluv má majitel přednastavenou šablonu v MS Word, v některých případech může být smlouva vytvořena odběratelem, případně stejný účel může splnit závazná objednávka prací zasláná odběratelem
- předávací protokol – dokument, který se předává odběrateli po ukončení prací a obsahuje prohlášení o správnosti provedených prací, datum předání atd.
- faktura – obsahuje konečnou částku fakturovanou odběrateli, která zahrnuje cenu spotřebovaného materiálu i cenu práce, tedy marži pro firmu, faktury majitel vytváří pomocí programu POHODA

Hlavním důvodem neefektivnosti tohoto způsobu evidence je především potřeba zpětného vyhledávání v datech zaznamenaných o zakázkách. Papírová databáze neumožňuje rychlé vyhledávání, zjištění celkového počtu zhotovených zakázek za určité období, celkový přehled spotřebovaného materiálu za určité období atd. Z těchto nedostatků vyplývá, že zavedení elektronické databáze vedené pomocí nějaké softwarové aplikace by jednoznačně zrychlilo a zefektivnilo pozdější využívání uložených dat.

## 2.3 Požadavky na databázi

Spolu s majitelem firmy bylo stanoveno několik základních požadavků, které by vytvořená databáze měla splňovat. První omezení se týká samotného softwarového nástroje, ve kterém bude vytvořená databáze implementovaná. Jelikož o správu celé databáze se po implementaci bude starat pouze majitel firmy sám, je potřebné zvolit program, který bude jednoduchý na ovládání, bude mít přívětivé uživatelské prostředí a propracovaný systém nápovědy, která majiteli usnadní pozdější zpracovávání dat.

Dále musí navrhovaná databáze obsahovat všechny informace týkající se jednotlivých zakázek provázané tak, aby bylo možné vyhledávat nad těmito daty. Mezi základní informace, které by měli být ve vytvořené databázi dohledatelné, patří:

- celkový počet zakázek uskutečněných za určité časové období (např.: čtvrtletí)
- čistý příjem za zakázky v určitém časovém období
- celkové náklady na zakázky v určitém časovém období
- celkový počet přijatých nabídek za určité časové období
- přehled pracovníků a jejich účast na jednotlivých zakázkách

Posledním konkrétním požadavkem stanoveným majitelem je prvotní zpracování starších dat firmy, určených k archivování. Ze zákona danou povinností je majitel povinen uchovávat informace o zakázkách minimálně tři roky zpětně, z tohoto důvodu se majitel chce v první řadě zaměřit na archivaci dat týkajících se zakázek provedených v letech 2009-2011.

### 3. Tvorba konceptuálního návrhu

V následující kapitole bude vytvořen konceptuální návrh tvořené databáze, tak aby co nejvěrněji kopíroval prostředí dané firmy a mohl být funkčním návrhem, který bude dále využit k tvorbě databáze. V této fázi je důležité správně vymezit entity a jejich vzájemné vztahy.

#### 3.1 Vymezení entit

Prvním krokem je nalezení všech entit, tedy objektů dané reality, které nás zajímají a o nichž chceme uchovávat informace. Podle analýzy prostředí (viz 2.2) a rozhovoru s majitelem firmy byly vytvořeny tyto entity:

- nabídka
- smlouva
- zakázka
- objednávka
- pracovník
- předávací protokol
- faktura
- dodavatel
- odběratel

Entita nabídka reprezentuje situaci, kdy odběratel poptává u firmy provedení nějakých prací, majitel pak na základě informací od odběratele vyhotovuje nabídku, ve které propočítává náklady na provedení veškerých prací i potřebný materiál a stanovuje tak odběrateli cenu za zhotovení zakázky. Odběratel nemusí vždy danou nabídku přijmout. Do databáze by měli být ukládány informace o všech majitelem vytvořených nabídkách, právě prostřednictvím této entity.

Dále by si majitel přál uchovávat informace o dokumentu, na jehož základě se daná zakázka začne realizovat, tento účel bude splňovat entita smlouva. Entita zakázka bude obsahovat souhrnné informace o místě a způsobu zpracování dané zakázky. Entita objednávka byla vytvořena zejména proto, aby bylo možné kontrolovat náklady vynaložené za materiál v rámci každé zakázky. Vzhledem k tomu, že pracovníci jsou na potřebné práce zvlášť nájímáni (viz 2.1) si majitel přeje zaznamenávat informace o jednotlivých pracovnících a jejich účasti na zakázkách.

Předávací protokol je dokument, který ukončuje každou zakázku a jeho podepsání oběma stranami vyjadřuje souhlas odběratele s provedenými pracemi na jím zadaném projektu a prohlášení majitele firmy o správnosti jím provedených prací. Každá ukončená zakázka musí být následně odběrateli vyfakturována, a informace o vystavených fakturách ponese entita faktura.

Informace o odběratelích a dodavatelích firmy jsou také nezbytnou součástí databáze, proto nesmí chybět tyto dvě entity pro uchování všech potřebných dat.

### 3.2 Přidělení atributů

Nyní je potřeba stanovit vlastnosti entit, pomocí nichž budou rozeznávány jednotlivé výskyty těchto entit. Přehled jednotlivých atributů každé entity je uveden v tabulce 2.

**Tabulka 2:** Entity a jejich atributy

Entita	Atributy
<b>Nabídka</b>	datum, odběratel, částka, druh práce, číslo nabídky
<b>Smlouva</b>	druh smlouvy, odběratel, místo, datum uzavření, termín dokončení, číslo smlouvy
<b>Zakázka</b>	název akce, druh práce, místo výkonu, počet pracovníků, číslo zakázky
<b>Objednávka</b>	datum, dodavatel, částka, počet položek, číslo objednávky
<b>Pracovník</b>	jméno, příjmení, adresa, telefon, kvalifikace
<b>Předávací protokol</b>	datum předání, číslo protokolu
<b>Faktura</b>	datum vystavení, datum splatnosti, skutečná splatnost, odběratel, částka, číslo faktury
<b>Dodavatel</b>	název, adresa, telefon, mail, IČO, DIČ
<b>Odběratel</b>	název, adresa, telefon, mail, IČO, DIČ

*Zdroj: vlastní zpracování*

Entita nabídka má jako atributy přiděleno datum vystavení příslušné nabídky, odběratele, kterému byla nabídka poskytnuta a konečná cena, za kterou je firma schopna danou práci vykonat. Dále majitel firmy vyslovil přání rozlišovat nabídky, které se týkají montáže suchých staveb a elektroinstalací, k tomuto účelu slouží atribut druh práce.

Atribut druh smlouvy u entity smlouva slouží k odlišení typu dokumentu, na jehož základě byla zakázka zahájena. Některé zakázky mají vystavenou smlouvu o dílo, jiné mohou být zahájeny pouze na základě přijaté objednávky od odběratele. Dále se u smlouvy budou zanášet informace o místě a datu uzavření a odběratelem požadovaný termín dokončení.

Ve firmě je zvykem jednotlivé zakázky pojmenovávat, aby se usnadnila pozdější kontrola dané akce, dále pak budou zakázky stejně jako nabídky rozlišovány podle druhu práce, v závislosti jedná-li se o suchou stavbu nebo elektro. Dále si majitel přeje zaznamenávat informace o místě, ve kterém zakázka probíhala a o celkovém počtu pracovníků, kteří se na pracích podíleli.

U entity objednávka jsou nejpodstatnějšími atributy dodavatel a částka. Předávací protokol se vytváří na konci zakázky a slouží hlavně jako doklad správného předání stavby zadavateli, nejdůležitějším atributem je pak datum, podle kterého se dá určit, jestli byla práce předána v požadovaném termínu.

U faktur se kromě klasických dat vystavení a splatnosti bude zaznamenávat i datum skutečné splatnosti, aby se zpětně dalo kontrolovat odběratelovo včasné splácení. Entity pracovník, odběratel a dodavatel obsahují klasické atributy pro odlišení jednotlivých entit. Majitel si navíc u každého pracovníka přeje uchovávat informace o tom, jakým oborem se daný člověk zabývá (zda je to např.: elektrikář, montér, zedník), k těmto účelům je přidán atribut kvalifikace.

### **Stanovení identifikátorů**

V následujícím kroku je potřebné stanovit mezi zvolenými atributy identifikátor neboli klíč, jehož hodnota bude jednoznačně odkazovat na daný výskyt entity.

Pro většinu entit byl jako identifikátor zvolen atribut s názvem číslo, z toho důvodu, že většinou jde o entity zastupující majitelem vytvářené dokumenty, které jsou při vytváření číslovány, a nenastane situace, že by dva dokumenty stejného typu měli stejné číslo, proto se tento atribut stává unikátním pro každý výskyt dané entity a je tudíž vhodné použít ho jako identifikátor.

U entit odběratel a dodavatel bylo za klíč zvoleno identifikační číslo (IČO), jelikož toto číslo má přidělené každý podnikatel i firma a je zaručena jeho jednoznačnost. Pro entitu pracovník byl jako identifikátor vytvořen nový atribut ID pracovníka, a to především z důvodu, že ani jeden stávající atribut nesplňuje podmínky pro klíč entity (viz 1.2.2).

Tabulka 3 zobrazuje finální podobu entit a jejich atributů i s identifikátory zvýrazněnými pomocí podtržení.

**Tabulka 3:** Entity, atributy a jejich identifikátory

Entita	Atributy
<b>Nabídka</b>	datum, odběratel, částka, druh práce, <u>číslo nabídky</u>
<b>Smlouva</b>	druh smlouvy, odběratel, místo, datum uzavření, termín dokončení, <u>číslo smlouvy</u>
<b>Zakázka</b>	název akce, druh práce, místo, počet pracovníků, <u>číslo zakázky</u>
<b>Objednávka</b>	datum, dodavatel, částka, počet položek, <u>číslo objednávky</u>
<b>Pracovník</b>	jméno, příjmení, adresa, telefon, kvalifikace, <u>ID pracovníka</u>
<b>Předávací protokol</b>	datum předání, <u>číslo protokolu</u>
<b>Faktura</b>	datum vystavení, datum splatnosti, skutečná splatnost, odběratel, částka, <u>číslo faktury</u>
<b>Dodavatel</b>	název, adresa, telefon, mail, <u>IČO</u> , DIČ
<b>Odběratel</b>	název, adresa, telefon, mail, <u>IČO</u> , DIČ

*Zdroj: vlastní zpracování*

### 3.3 Vztahy mezi entitami

V následující podkapitole budou stanoveny vztahy mezi jednotlivými entitami. U vztahů bude určena i kardinalita a parcialita, jako integritní omezení vztahu, která zajistí správnost dat. Popisované vztahy jsou zachycené na výsledném E-R diagramu (obrázek 4), každý vztah je zvýrazněn jinou barvou pro lepší orientaci při následném popisu.

První vztah vniká mezi entitami nabídka a odběratel, tento vztah byl pojmenován se nabízí a v diagramu je označen červenou barvou. Vyjadřuje situaci, kdy každá konkrétní nabídka je nabídnuta pouze jedinému odběrateli. Tento fakt vyjadřuje v použité MIN-MAX notaci označení 1,1 na levé straně vztahu. Při určování kardinality a parciality je nutné podívat se na vztah i z pohledu druhé entity, zkoumaný vztah má tedy z pohledu odběratele toto vyjádření: jeden odběratel musí dostat od naší firmy nabídnutou jednu nebo i více nabídek. To je dáno firemní realitou, kdy jedna firma může opakovaně žádat o vykonání různých prací. Po pravé straně vztahu tuto situaci zastupuje symbol 1,N.

Další vztah vzniká ve chvíli, kdy dá odběratel pokyn k zahájení dané zakázky. Ze strany odběratele je pak vztah popsán tak, že jeden odběratel může zadat vykonání jedné nebo více zakázek, ale z pohledu druhé entity musí být každá zakázka zadána maximálně jedním odběratelem. V celkovém diagramu je tento vztah nazván zadává a zvýrazněn je zelenou barvou.

Každá zakázka pak musí být započatá na základě maximálně jedné smlouvy, to znamená jedním druhem smlouvy podle rozlišení, které je popsáno v předešlé kapitole a ve které je uvedeno přesné datum ukončení prací podle odběratele. Tento vztah byl nazván se začíná a v diagramu je označen modrou barvou.

Na každou zakázku je pak potřeba najmout určitý počet pracovníků. Konečný počet je závislý na charakteru vykonávané práce, vždy však musí být najatý minimálně jeden pracovník. Z pohledu pracovníka pak můžeme žlutý vztah najmutí charakterizovat takto: daný pracovník může, ale nemusí být najatý na jednu i více zakázek. Tento vztah je dále charakterizován několika atributy. Majitel firmy si přál uchovávat informace o datu najmutí jednotlivých pracovníků, o celkovém počtu dní, které na zakázce pracovník odpracoval, v jakém pracovním poměru byl na zakázku najat a jaká částka mu byla za provedenou činnost zaplácena.

Z rozhovoru s majitelem dále vyplynulo, že na každou zakázku je potřebné objednat materiál pomocí objednávky. Toto reprezentuje fialový vztah zhotovení, který říká, že pro každou zakázku musí být zhotovena alespoň jedna objednávka, ale z druhého pohledu se konkrétní objednávka musí vztahovat maximálně k jedné zakázce.

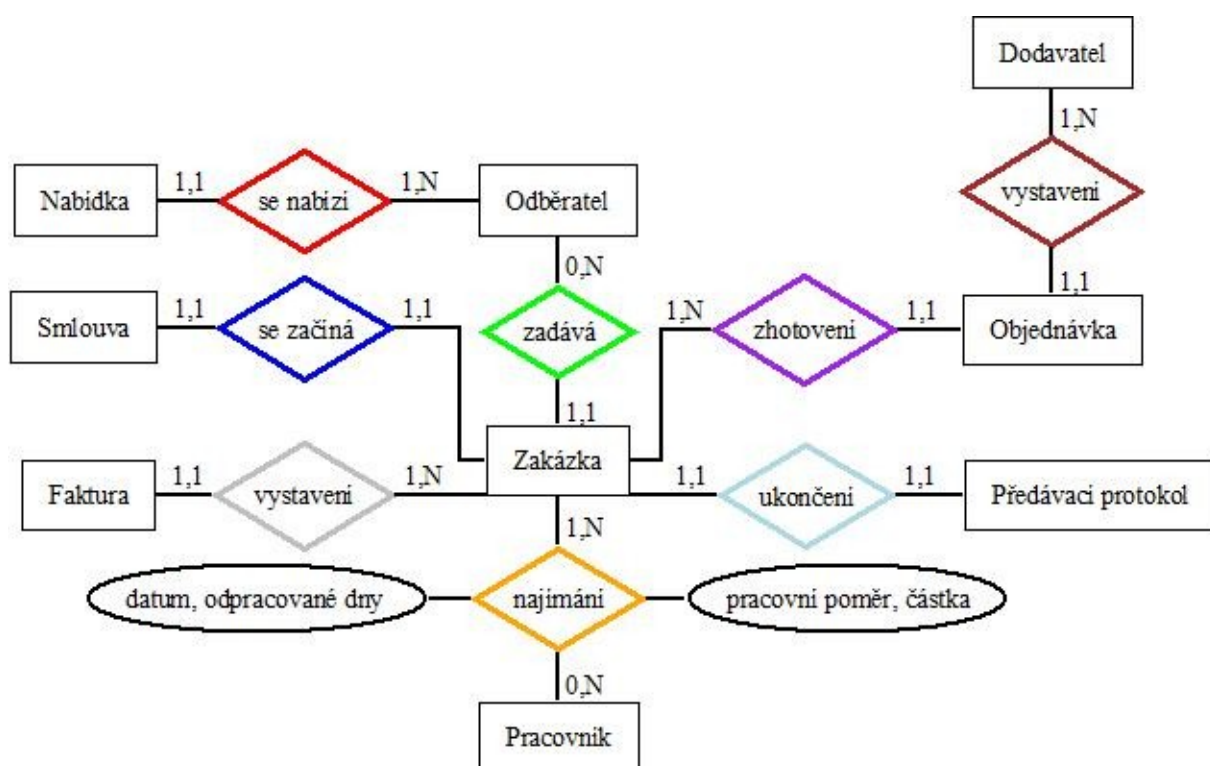
Tato objednávka pak musí být vystavena konkrétnímu dodavateli materiálu, který musí objednávku zpracovat. Jeden dodavatel však může zpracovávat více objednávek, například z různých zakázek. Tento vztah byl nazván jako vystavení a je označen hnědou barvou.

Po zakončení prací se zakázka ukončuje předáním a podepsáním předávacího protokolu majitelem firmy a zástupcem odběratele. Vztah mezi zakázkou a předávacím protokolem byl nazván ukončení, v diagramu je označen světle modrou barvou a lze ho popsat takto: každá zakázka musí být ukončena jedním předávacím protokolem a zároveň každý předávací protokol musí ukončovat pouze jednu konkrétní zakázku.



Poslední vznikající vztah je mezi zakázkou a fakturou, jelikož za každou zakázku si firma fakturuje za spotřebovaný materiál a provedenou práci. Z pohledu faktury je vztah definován symbolem 1,1 což vyjadřuje fakt, že každá faktura se vztahuje pouze k jedné zakázce. V praxi je však naprosto běžné, že například větší zakázky se fakturují postupně již během zhotovování, proto je z pohledu zakázky použit symbol 1,N, který vyjadřuje nutnost účasti entity zakázka na vztahu, ale i s více než jedním výskytem. Tento vztah vystavení je v diagramu označen šedivou barvou.

Všechny výše popsané vztahy dohromady tvoří E-R diagram modelované reálie, který je zobrazen na následujícím obrázku 4. Tento diagram bude následně transformován do relačního modelu dat.



**Obrázek 4:** E-R diagram modelované reálie

*Zdroj: vlastní zpracování*

## 4. Tvorba relačního modelu dat

V této kapitole bude provedena transformace E-R diagramu vytvořeného v předcházející kapitole. Nejprve bude provedena transformace entit a vztahů mezi nimi, vzniklé relace budou následně normalizovány podle normálních forem.

### 4.1 Návrh relací

V předcházejících kapitolách byl vytvořen konceptuální model, který bude nyní transformován do relačního modelu dat. Transformace bude prováděna po jednotlivých vztazích a použita bude teorie z kapitoly 1.4.



**Obrázek 5:** Vztah Nabídka – Odběratel.

*Zdroj: vlastní zpracování*

Na obrázku 5 je zobrazen první vztah mezi nabídkou a odběratelem. Dochází k transformaci vztahu 1:N. Vzhledem k povinnému členství obou entit vzniknou dvě relace, nabídka a odběratel. Primárními klíči se stanou původní identifikátory entit, tedy číslo nabídky a IČO odběratele. Aby mohli být tyto dvě relace spojeny do vzájemného vztahu, dostane relace nabídka jako determinant vztahu přiděleno IČO odběratele jako cizí klíč.



**Obrázek 6:** Vztah Zakázka – Odběratel.

*Zdroj: vlastní zpracování*

Další vztah (obrázek 6) mezi zakázkou a odběratelem je opět typu 1:N a i v tomto případě vzniknou dvě relace. Nová relace zakázka a relace odběratel, která je totožná s relací, která vznikla v předchozím kroku. Relace zakázka je opět determinantem vztahu a jako cizí klíč jí bude přiděleno IČO odběratele. Primárním klíčem determinantu se stane číslo zakázky.



**Obrázek 7:** Vztah Zakázka – Smlouva.

*Zdroj: Vlastní zpracování*

Obrázek 7 zachycuje vztah entit zakázka a smlouva. Obě tyto entity mají ve vztahu povinný maximálně jeden výskyt, z čehož vyplývá, že vznikne pouze jedna relace, která bude obsahovat atributy obou entit. Jako praktické řešení se jeví ponechat relace zakázka, které již vznikla v předchozím kroku a přidat k ní všechny atributy původní entity smlouva. Primárním klíčem zůstane číslo zakázky.



**Obrázek 8:** Vztah Zakázka – Faktura.

*Zdroj: vlastní zpracování*

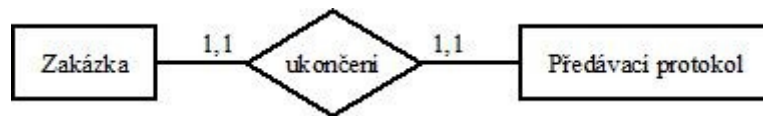
Dalším transformovaným vztahem je vztah 1:N entit zakázka a faktura (obrázek 8). Transformací vzniknou opět dvě relace, již existující zakázka a nově vzniklá faktura, která bude jako cizí klíč obsahovat číslo zakázky. Primárním klíčem relace faktura bude číslo faktury.



**Obrázek 9:** Vztah Zakázka – Pracovník.

*Zdroj: vlastní zpracování*

Vztah zakázka a pracovník (obrázek 9) je jediným vztahem typu M:N v modelu. Reprezentací tohoto vztahu vzniknou tři relace. Relace zakázka, která již byla vytvořena, relace pracovník s atributy původní entity a primárním klíčem ID pracovníka a relace vztahu. Tato relace bude pojmenována najmutí pracovníka a bude obsahovat číslo zakázky a ID pracovníka jako cizí klíče. Tyto oba klíče budou zároveň tvořit složený primární klíč této relace. Podle obrázku 4 má také tento vztah nějaké atributy, které budou při transformaci také přiděleny této relaci.



**Obrázek 10:** Vztah Zakázka – Předávací protokol.

*Zdroj: vlastní zpracování*

Vztah na obrázku 10 představuje další vztah typu 1:1. I v tomto případě vznikne pouze jedna relace. Opět to bude relace zakázka s primárním klíčem číslo zakázky a budou jí přiděleny všechny atributy entity předávací protokol.



**Obrázek 11:** Vztah Zakázka – Objednávka.

*Zdroj: vlastní zpracování*

Předposledním vztahem je opět vztah typu 1:N (obrázek 11). Povinná účast determinantu ve vztahu napovídá vzniku dvou relací, pro každou z entit. Vznikne tedy nová relace objednávka, jejímž primárním klíčem bude číslo objednávky a bude obsahovat atributy entity. Aby mohlo vzniknout propojení těchto dvou relací, bude relaci objednávka přidán cizí klíč číslo zakázky.



**Obrázek 12:** Vztah Dodavatel – Objednávka.

*Zdroj: vlastní zpracování*

Posledním vztahem, který bude transformován, je vztah entit dodavatel a objednávka (obrázek 12). Jedná se opět o vztah 1:N s povinným členstvím determinantu ve vztahu. Transformací vzniknou dvě relace, existující objednávka a dodavatel. Primárním klíčem relace dodavatel se stane IČO dodavatele. K propojení obou relací se přidá toto IČO jako cizí klíč k relaci objednávka.

V následující tabulce 4 je přehled všech vzniklých relací, názvy jejich atributů a primární klíče, které jsou vyznačeny pomocí podtržení. Cizí klíče v relacích jsou označeny pomocí zkratky CK.

**Tabulka 4:** Přehled relací a názvů atributů

Relace	Názvy atributů
<b>Odběratel</b>	<u>IČO</u> , DIČ, název, adresa, telefon, mail
<b>Nabídka</b>	<u>číslo nabídky</u> , datum, částka, druh práce, IČO odběratele CK
<b>Zakázka</b>	<u>číslo zakázky</u> , název akce, druh práce, místo, počet pracovníků, číslo smlouvy, datum uzavření, termín dokončení, druh smlouvy, místo uzavření, číslo protokolu, datum předání, IČO odběratele CK
<b>Faktura</b>	<u>číslo faktury</u> , datum vystavení, datum splatnosti, skutečná splatnost, částka, číslo zakázky CK
<b>Pracovník</b>	<u>ID pracovníka</u> , jméno, příjmení, adresa, telefon, kvalifikace
<b>Najmutí pracovníka</b>	<u>číslo zakázky</u> , <u>ID pracovníka</u> , datum, odpracované dny, pracovní poměr, částka
<b>Objednávka</b>	<u>číslo objednávky</u> , datum, částka, počet položek, číslo zakázky CK, IČO dodavatele CK
<b>Dodavatel</b>	<u>IČO</u> , DIČ, název, adresa, telefon, mail

*Zdroj: vlastní zpracování*

## 4.2 Užití normálních forem

Aby se zajistila co nejnižší redundance a duplicita ukládaných dat do relací a zároveň se zvýšila spolehlivost vytvořeného datového modelu, budou relace testovány na normální formy (viz 1.5).

První normální forma (1NF) říká, že každý atribut v relaci musí být atomický, tedy dále nedělitelný. To znamená, že každý atribut nesmí obsahovat více než jednu hodnotu. Při pohledu na relace v tabulce 4 je patrné, že tuto podmínku nesplňuje atribut adresa, protože adresa obsahuje ulici, název města a PSČ.

Jedním řešením by bylo vytvoření nové relace adresa, která by obsahovala zmíněné atributy, primárním klíčem by mohlo být například PSČ přes které by byly obě relace propojené. V tomto případě však bude praktičtější odstranit z relace atribut adresa a nahradit ho přímo třemi novými atributy: ulice, město a PSČ. Atribut adresa obsahují tři relace: odběratel, pracovník a dodavatel, zmíněná změna bude provedena u všech (viz tabulka 5). Všechny ostatní relace danou podmínku splňují.

**Tabulka 5:** Relace upravené podle 1NF

Relace	Názvy atributů
<b>Odběratel</b>	<u>IČO</u> , DIČ, název, ulice, město, PSČ, telefon, mail
<b>Pracovník</b>	<u>ID pracovníka</u> , jméno, příjmení, ulice, město, PSČ, telefon, kvalifikace
<b>Dodavatel</b>	<u>IČO</u> , DIČ, název, ulice, město, PSČ, telefon, mail

*Zdroj: vlastní zpracování*

Aby relace splňoval druhou normální formu (2NF) musí platit, že splňuje 1NF a každý neklíčový atribut je plně funkčně závislý na celém primárním klíči. Tato forma se vztahuje pouze na relace, které mají primární klíč složený z více než jednoho atributu. Ve vytvořených relacích tuto podmínku splňuje relace najmutí pracovníka. Při pohledu na atributy, které tato relace obsahuje (datum najmutí, odpracované dny, pracovní poměr, částka) je zřejmé, že relace splňuje 1NF. Testování na 2NF pak proběhlo takto:

- datum najmutí – tento atribut je funkčně závislý na ID pracovníka, z toho důvodu, že každý pracovník může na potřebné práce nastoupit v různé dny, ale zároveň záleží na konkrétní zakázce, aby den nástupu odpovídal potřebným úkonům
- odpracované dny – každý pracovník může odpracovat různý počet dní, vzhledem k tomu, že jeden pracovník může být najímán opakovaně na jiné práce v rámci různých zakázek, je atribut závislý i na čísle zakázky
- pracovní poměr – tento atribut uchovává informace o pracovních poměrech, na jejichž základě pracovník práce vykonával (pracovní smlouva, práce na dohodu, brigáda...), každý pracovník může na různých zakázkách pracovat na základě jiného pracovního vztahu, proto je atribut závislý jak na konkrétním pracovníkovi, tak na dané zakázce

- částka – tento atribut představuje sumu, které byla pracovníkovi za jeho služby zaplácena, atribut je tedy závislý nejen na konkrétním pracovníkovi, ale i na zakázce a jejím celkovém rozpočtu.

Z výše uvedeného vyplývá, že relace splňuje podmínky 2NF a není tedy potřeba provádět nějaké úpravy.

Relace je ve třetí normální formě (3NF) tehdy, je – li v 1NF a 2 NF a žádný z atributů není tranzitivně závislý na klíči. To znamená, že každý atribut musí být závislý pouze čistě na primárním klíči a ne na žádném jiném neklíčovém atributu. Při zkoumání relací z tabulky 4 bylo zjištěno několik atributů, u nichž by mohla jistá závislost na jiném než primárním klíči existovat.

Prvním takovým případem jsou atributy datum uzavření, termín dokončení, druh smlouvy, místo uzavření z relace zakázka. Všechny tyto atributy obsahují informace, které jsou uvedené vždy v daném dokumentu ještě před zahájením zakázky. Dá se tady říci, že se vztahují spíše ke konkrétní smlouvě (ta je v relaci reprezentována atributem číslo smlouvy), než k prováděné zakázce. Řešením tohoto rozporu je vytvoření nové relace smlouva, která bude obsahovat všechny uvedené atributy a atribut číslo smlouvy jako primární klíč. Tyto atributy budou zároveň vyjmuty z relace zakázka, zůstane pouze číslo smlouvy jako cizí klíč, který umožní propojení obou relací.

Dalšími spornými atributy jsou PSČ a město. Vzhledem k faktu, že každé město má přidělené své jedinečné PSČ, dalo by se říci, že tento atribut je tedy závislý pouze na atributu město, což není primární klíč schématu. Tato situace bude opět řešena rozkladem na více relací. Vznikne tedy nová relace město, která bude obsahovat atributy PSČ a název města. Klíčem schématu se stane PSČ a bude ponecháno i ve všech původních relacích jako cizí klíč. Toto řešení má i své praktické opodstatnění. Touto relací vznikne „seznam“ všech možných měst a vyplňování pouze PSČ u ostatních relací podstatně ulehčí práci správci databáze.

Jistá závislost existuje i mezi atributy částka a odpracované dny v relaci najmutí pracovníka. V praxi samozřejmě závisí výše odměny na celkovém počtu odpracovaných dní (hodin), teoreticky by tedy měla být ošetřena i tato závislost. Pro majitele je však podstatné zaznamenávat pouze celkovou částku, která byla pracovníkovi vyplácena bez ohledu na to, jakým způsobem byla výše částky určena. Vytvoření nové relace by tedy nemělo žádný praktický význam a naopak by mohlo způsobit nepřehlednost a zbytečnou práci navíc.

Žádné jiné atributy již nejeví známky tranzitivní závislosti, nebudou tedy prováděny jiné úpravy, než ty které jsou uvedeny výše. Všechny upravené a nově vzniklé relace jsou znázorněny v tabulce 6.

**Tabulka 6:** Relace upravené podle 3NF

Relace	Názvy atributů
<b>Odběratel</b>	<u>IČO</u> , DIČ, název, ulice, PSČ CK, telefon, mail
<b>Pracovník</b>	<u>ID pracovníka</u> , jméno, příjmení, ulice, PSČ CK, telefon, kvalifikace
<b>Dodavatel</b>	<u>IČO</u> , DIČ, název, ulice, PSČ CK, telefon, mail
<b>Zakázka</b>	<u>číslo zakázky</u> , název akce, druh práce, místo, počet pracovníků, číslo smlouvy CK, číslo protokolu, datum předání, IČO odběratele CK
<b>Smlouva</b>	<u>číslo smlouvy</u> , datum uzavření, termín dokončení, druh smlouvy, místo uzavření
<b>Město</b>	<u>PSČ</u> , název města

*Zdroj: vlastní zpracování*

Výsledkem transformace a normalizace je tedy deset relací, které jsou připravené na implementování do nějakého konkrétního softwarového nástroje.



## 5. Implementace relačního modelu dat

V následující kapitole bude provedena implementace relací vzniklých v předchozí části práce. Před zahájením implementace je však potřebné zvolit vhodný softwarový program, ve kterém bude databáze vytvořena a data dále zpracovávána.

### 5.1 Výběr softwarového nástroje

V současné době je na trhu mnoho firem nabízejících různá softwarová řešení, která umožňují zpracování relačního modelu dat. Mezi konkrétní nástroje patří například Oracle Database nebo MySQL.

S přihlédnutím k požadavkům na databázi definovaných majitelem, bude k implementaci databáze využit program, který je snadno dostupný, uživatelsky přívětivý, s přehlednou a funkční nápovědou. Vzhledem k tomu, že navrženou databázi bude spravovat a využívat pouze majitel a nikdo jiný, je zbytečné využívat databázi, které je umístěná na serveru a poskytuje tak možnost sdílení dat s více uživateli. Jako optimální programové řešení byl tedy zvolen Microsoft Access.

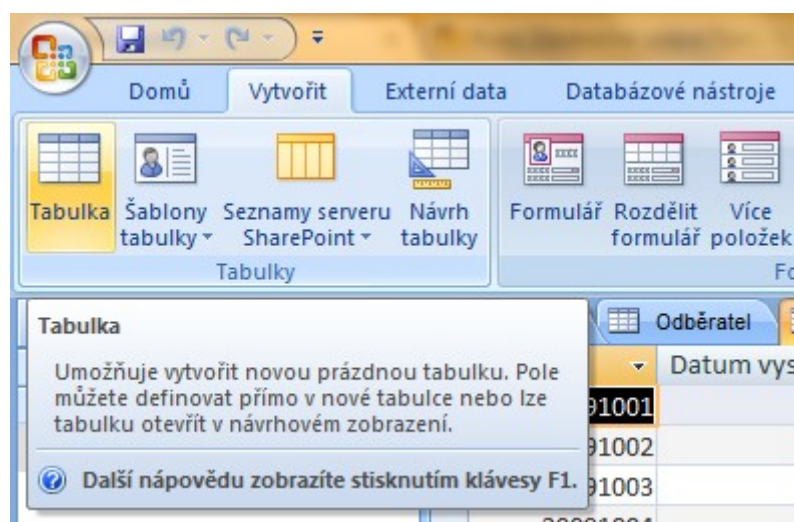
#### 5.1.1 Microsoft Access 2007

Access je jednou ze součástí celkové strategie Microsoftu v oblasti produktů pro správu dat. Kromě klasického propojení souvisejících uložených informací nabízí i další doplňující funkce. Umožňuje pracovat s daty z jiných zdrojů, například i s databázi uložených na SQL serverech. Plně také podporuje technologii Microsoftu OLE (Object Linking and Embedding), přičemž může vystupovat jako klient i server pro ostatní aplikace Microsoft, například Word, Excel, PowerPoint. Access také nabízí propracovaný systém na vývoj aplikací pro operační systém Microsoft Windows, pomocí něhož je možné vytvářet aplikace nezávislé na datovém zdroji. Pro malé firmy tedy Access představuje vše, co je potřeba pro ukládání a správu dat, využívaných pro běh podniku.[16]

V závislosti na faktu, že majitel již používá ostatní aplikace Microsoftu (Word, Excel) je pro něj uživatelské prostředí tohoto programu přijatelné a samotný program má již nainstalovaný, jako součást balíčku Microsoft Office.

## 5.2 Tvorba tabulek

Po vytvoření nové databáze v programu Access je prvním krokem implementace vytvořeného modelu dat tvorba jednotlivých tabulek, které odpovídají vytvořeným relacím. V práci byla využita tvorba tabulek pomocí záložky Vytvořit → Tabulka (viz obrázek 13). Samotné přidávání jednotlivých sloupců, které odpovídají atributům relací, probíhalo pomocí návrhového zobrazení nově vytvořené tabulky.



Obrázek 13: Vytvoření nové tabulky

*Zdroj: vlastní zpracování*

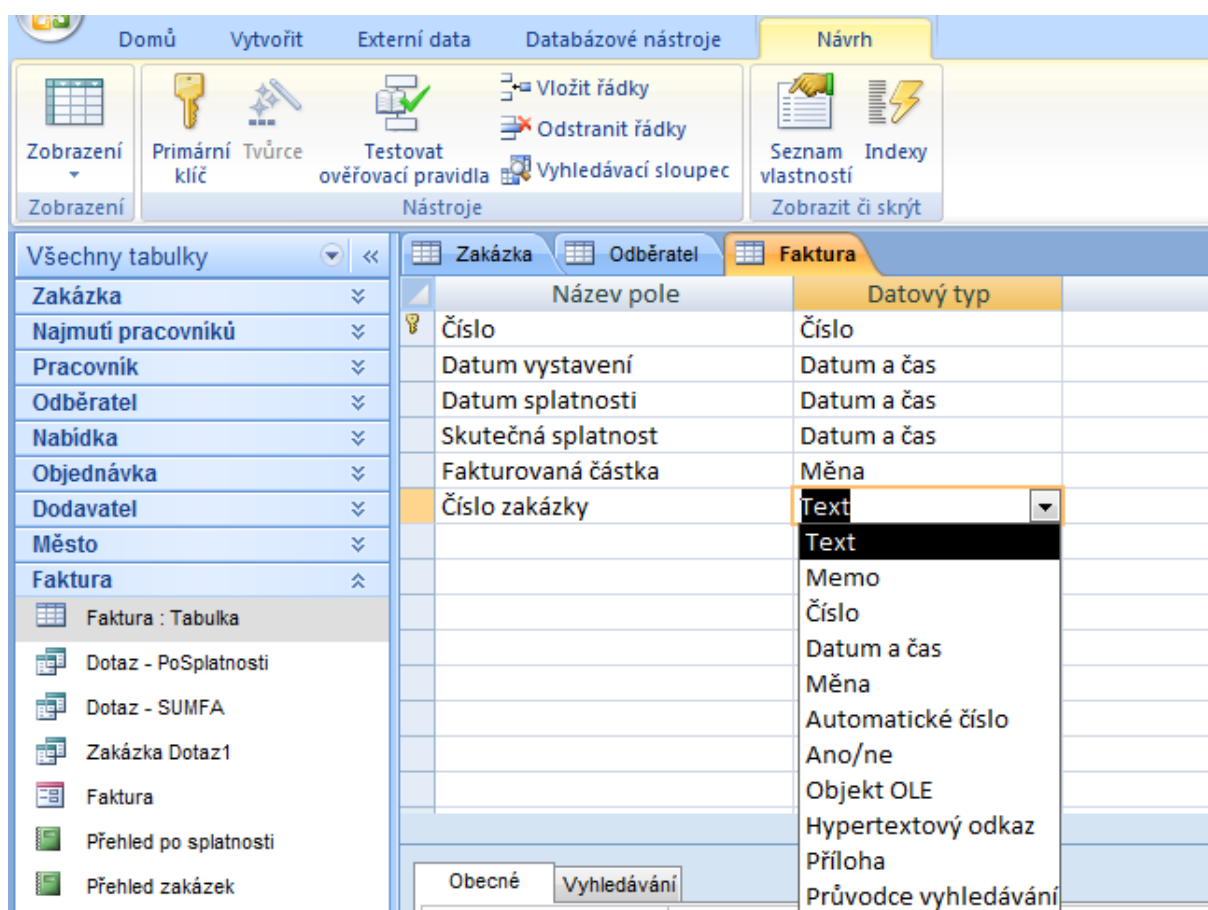
### 5.2.1 Definování polí

Každému atributu bude tedy odpovídat příslušné pole dané tabulky. V návrhovém zobrazení tabulky lze potom každému poli nastavit příslušný datový typ, aby bylo zajištěno pozdější ukládání správných dat. Dle [10] Microsoft Access 2007 nabízí tyto datové typy:

- Příloha – soubory, jako jsou například digitální fotografie, k záznamu může být přiřazeno více souborů, tento datový typ není k dispozici v předchozích verzích aplikace Access
- Automatické číslo – čísla, která se automaticky generují pro jednotlivé záznamy
- Měna – peněžní hodnoty
- Datum a čas – kalendářní data a časy
- Hypertextový odkaz – hypertextové odkazy, jako jsou e-mailové adresy
- Memo – dlouhé bloky textu a text s formátováním, typické použití pole typu Memo je podrobný popis produktu

- Číslo – číselné hodnoty, například vzdálenosti
- Objekt OLE – Objekty OLE, jako jsou dokumenty aplikace Word.
- Text – krátké alfanumerické hodnoty, jako je příjmení nebo adresa.
- Ano/Ne – logické hodnoty
- Průvodce vyhledáváním – pomáhá vytvořit vyhledávací pole, které buď zobrazí seznam hodnot, které se načítají z tabulky či dotazu, nebo statistickou sadu hodnot, která byla zadaná při vytváření pole

Při tvorbě tabulek v této práci byly používány datové typy text, číslo, měna, datum a čas a průvodce vyhledáváním. Podoba návrhového zobrazení, vytváření jednotlivých tabulek a přiřazování datových typů polím je zobrazeno na obrázku 14.



**Obrázek 14:** Návrhové zobrazení a datové typy

*Zdroj: vlastní zpracování*

Každé pole pak může být dále charakterizováno různými vlastnostmi, aby se co nejvíce ošetřila podoba později vkládaných dat. Vlastnosti pole jsou závislé na datovém typu, kde každý typ může být dále specifikován různými vlastnostmi, které upravují například to, kolik znaků může mít hodnota ukládaná do pole, jaký bude formát ukládané hodnoty, u čísel může

být specifikován počet desetinných míst apod. Při tvorbě polí pro tuto práci byly nejdůležitější tyto vlastnosti:

- Je nutno zadat – tuto vlastnost lze definovat u každého datového typu a zajišťuje vynucení toho, aby každý záznam měl v tomto poli vloženou hodnotu, tato vlastnost byla nastavena u všech polí, které představují cizí klíče v tabulce, aby byla zajištěna konzistentnost databáze
- Indexovat – vlastnost zajistí vytvoření indexu pole, což urychlí přístup pro čtení dat v tomto poli, indexování je nastaveno vždy u primárního klíče tabulky

### 5.2.2 Definování relací

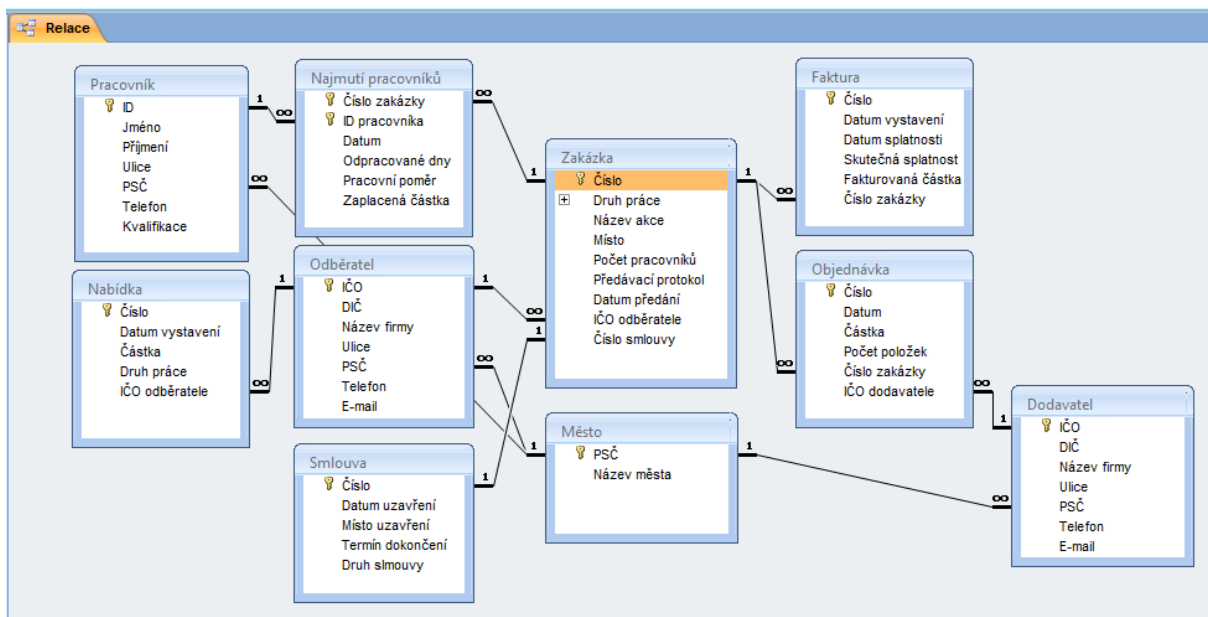
Po vytvoření všech tabulek je důležité nastavit vztahy mezi těmito tabulkami. Prakticky se jedná o původní vztahy identifikované mezi entitami na konceptuální úrovni, dále upravené při transformaci na relační model dat. Tyto vztahy se v programu Access nazývají relace.

Z pohledu programu pak existují tyto důvody, proč je nezbytné relace mezi tabulkami definovat:

- relace mezi tabulkami informují návrhy dotazů – důležité při tvorbě dotazů, které mají vrátit informace z více tabulek
- relace mezi tabulkami informují návrhy formulářů a sestav – i formuláře a sestavy mohou být vytvořeny z dat v různých tabulkách
- relace mezi tabulkami jsou základnou pro vynucení referenční integrity – pomocí referenční integrity lze zabránit vzniku osiřelých záznamů, tedy záznamů, které mají referenci na neexistující záznam [8]

Program opět nabízí několik způsobů vytvoření relací mezi tabulkami, v práci byl využit způsob definování pomocí karty Databázové nástroje → Vztahy. Relace jsou pak opět vytvářeny mezi primárními klíči a odpovídajícími cizími klíči ve druhé tabulce. U každé relace bylo nastaveno zajištění referenční integrity.

Vzniklé relace mezi vytvořenými tabulkami v programu jsou zobrazeny na obrázku 15. Zajištění referenční integrity u relace znázorňuje tučně zvýraznění konců relací. Zobrazují se i typy jednotlivých relací. Relace typu  $1,\infty$  reprezentuje původní vztah 1:N.



**Obrázek 15:** Relace mezi tabulkami

*Zdroj: vlastní zpracování*

### 5.3 Vkládání dat

Po vytvoření všech tabulek a jejich propojení pomocí relací je možné začít databázi plnit daty. Program Access mimo jiné nabízí i možnost automatického importu dat z různých formátů, jelikož ovšem majitel nemá žádný informační systém, ani žádným způsobem osvojený jednotný způsob vytváření příslušných dokumentů obsahující potřebná data, musel být zvolen ruční způsob nahrávání dat. Data byla nahrávána přímo v zobrazení datového listu příslušné tabulky.

Jelikož jedním z požadavků majitele bylo prvotní zpracování starších dat určených k archivaci, začaly se do databáze jako první ukládat data týkající se zakázek z roku 2009. Hned v prvním kroku seznamování se s daty určenými k ukládání do databáze, se ukázalo, jak nedokonalá a nepřehledná je papírová archivace.

Majitel se snažil zakládat dokumenty týkající se každé zakázky do jednotlivých složek a šanonů. Jelikož je ovšem tato práce poměrně časově náročná, nechával ji vždy zpravidla na konec roku, kdy následně zpětně třídil různé dokumenty a vytvářel z nich jednotlivé složky zakázek. Tento způsob je samozřejmě velmi neefektivní, kromě velké ztráty času dochází i k neúplnosti vytvořené databáze, jelikož ne vždy se potřebné dokumenty daly dohledat. Vzhledem k těmto okolnostem vzniklo několik problémů při ukládání dat.

První problém nastal u informací o nabídkách. Vzhledem k tomu, že majitele během roku oslovují různí odběratelé s nabídkami práce, dalo by se říci, že majitelem vytvořených nabídek by mělo být více než skutečně zhotovených zakázek. Majitel ovšem zpracovával nabídky elektronicky prostřednictvím MS Excel a způsob ukládání vytvořených nabídek byl poměrně zmatený. Někdy se stávalo, že majitel jako šablonu použil již existující nabídku, například pro stejného odběratele, ovšem bez toho, aby byla původní nabídka uložena. Navíc v polovině roku 2010 došlo k poškození počítače, na kterém majitel ukládal zhotovené nabídky a kvůli neexistujícímu zálohování, byla ztracena převážná část všech elektronických dat firmy. Vzhledem k těmto okolnostem nejsou data o nabídkách uložené v databázi zcela kompletní a takové formě, v jaké se původně předpokládalo.

Dalším problémem se stalo ukládání dat o objednávkách materiálu. V průběhu modelování a vytváření prvotního návrhu databáze se z rozhovorů s majitelem jevílo jako samozřejmé, že má zavedený nějaký systém na objednávání materiálu od dodavatelů. Při studiu dostupných dat o zakázkách se ukázalo, že bohužel neexistuje žádný systém a majitel potřebný materiál objednává vždy podle potřeby a to různými způsoby (telefonicky, e-mailem, osobním odběrem). K dispozici tedy byly různé typy dokumentů, jako dodací listy, paragony, přijaté faktury apod., navíc s naprostou nemožností přiřazení jednotlivých dokumentů ke konkrétním zakázkám.

Po rozhovoru s majitelem se ukázalo, že objednávku od začátku uváděl z toho důvodu, že by rád do budoucna měl nějaký konkrétní přehled o materiálních nákladech na jednotlivé zakázky. S tabulkami objednávka a dodavatel bude tedy počítáno při návrhu opatření pro budoucí správu databáze, ale v této fázi ukládání dat není možné tak rozsáhlé a nezpracované informace do databáze zahrnout, proto budou tabulky ponechány bez dat.

Data do ostatních tabulek byla vyplněna do takové míry, do jaké byly zjistitelné původní informace v dokumentech. Nikde jinde se však již neobjevil žádný rozsáhlejší problém s daty. Do databáze tedy bylo nahráno celkem 48 zakázek za rok 2009, 51 za rok 2010 a 45 zakázek za rok 2011. Dále byly zaznamenány informace o celkem 29 pracovnících, kteří se podíleli na zakázkách během těchto tří let a 70 odběratelů, pro které byly zakázky zhotoveny. Vzhledem k problémům uvedeným v této kapitole byly získány informace pouze o 63 nabídkách. Největší počet záznamů obsahuje tabulka faktura, která nese informace o celkem 216 vydaných fakturách.

## 5.4 Formuláře, dotazy, sestavy

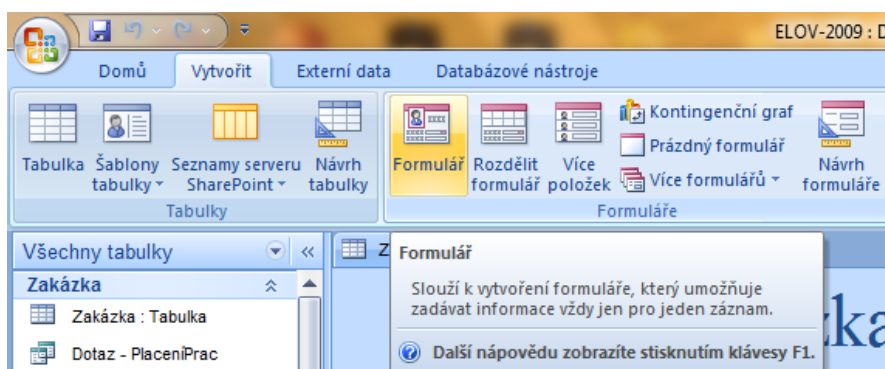
Po vložení všech dostupných dat byly provedeny i některé operace, které poslouží jako přehled majiteli o stavu firmy a zároveň prověří konzistentnost vytvořené databáze. Ověří se tedy to, zda je možné z takto navrženého modelu získávat potřebná data.

### 5.4.1 Formuláře

Formuláře většinou tvoří první rozhraní mezi uživatelem a aplikací a můžou být navrženy k mnoha různým účelům:

- zobrazení a úprava dat – nejčastější způsob využití formulářů, umožňují uživatelsky přizpůsobit podání dat z databáze, lze provádět i úpravu, vkládání nebo odstranění dat
- řízení toku aplikace – mohou pracovat s makry nebo s procedurami Microsoft Visual Basicu for Applications
- příjem vstupů – formuláře pro vstup nových dat
- zobrazování hlášení – mohou nabízet informace o tom, jak používat aplikaci, nebo o nastávajících činnostech
- tisk informací [16]

V databázi v rámci této práce byly vytvořeny formuláře, které budou sloužit ke vkládání nových údajů do tabulek. Majitel firmy se při tvorbě vyjádřil, že právě zobrazení dat pomocí formuláře je pro něj přirozenější a vkládání nových dat mu formuláře usnadní. Proto byly vytvořeny jednoduché formuláře pro každou tabulku, tak aby se co nejvíce usnadnilo pozdější vkládání a úprava dat. Formuláře byly vytvořeny pomocí nástroje Formulář na kartě Vytvoření (viz obrázek 16). Ukázka vytvořeného formuláře pro vkládání dat je v příloze A.



Obrázek 16: Vytvoření nového formuláře

*Zdroj: vlastní zpracování*

## 5.4.2 Dotazy

Dotaz představuje požadavek na zobrazení výsledků dat, na provedení akce s daty nebo na kombinaci dvou předchozích operací. Pomocí dotazů mohou být zodpovězeny jednoduché otázky, provedeny výpočty, data sloučena z různých tabulek nebo lze dokonce data přidat, změnit nebo odstranit z tabulky. Program Access používá dva druhy dotazů:

- výběrové dotazy – pro načtení dat z tabulek nebo pro výpočty
- akční dotazy – pro přidávání, změnu nebo odstranění dat [9]

Nad daty ve vytvořené databázi byly prováděny klasické výběrové dotazy, které spojují informace z různých tabulek tak, aby vznikl ucelený přehled o určitém období ve firmě. Dotazy byly tvořeny hlavně na základě požadavků stanovených v kapitole 2.3.

Nesplněné zůstaly pouze požadavky na celkové náklady zakázky a celkový počet nabídek za určité období. Dotazy na získání těchto informací nemohly být vytvořeny z důvodu absence potřebných dat v databázi. Vytvořeny tedy byly dotazy na zjištění celkového počtu zakázek za jednotlivé roky (2009-2011), pro každý rok byl proveden součet fakturovaných částek, jako náklady byly pouze sečteny částky vyplacené zaměstnancům. Dále byly vytvořeny dotazy na zjištění zakázek, které byly předány po termínu předání stanoveném odběratelem a faktury, které byly zaplacené po splatnosti.

Veškeré vytvořené dotazy sloužily jako podklad pro následné vytvoření přehledových sestav a byly vytvořeny pomocí karty Vytvoření → Průvodce dotazem. Ukázka návrhového zobrazení dotazů je v příloze B.

## 5.4.3 Sestavy

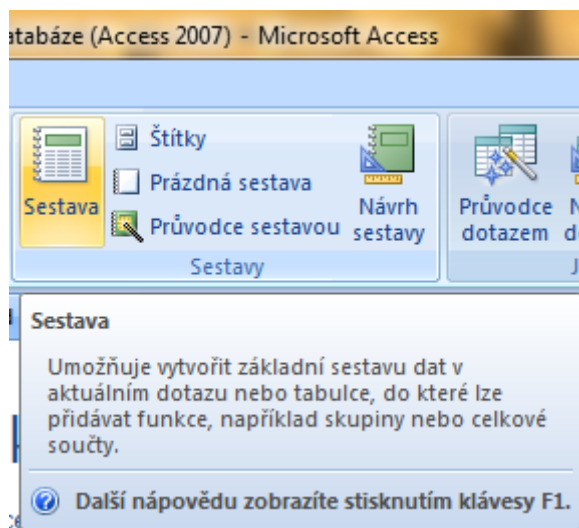
Sestavy představují nejlepší způsob pro vytváření tištěných kopií informací, které jsou vybírány nebo vypočítány z uložených dat v databázi. Mají dvě hlavní výhody před jinými způsoby tisku:

- mohou provádět srovnání, vytvářet souhrnné výpočty a podsoučty nad velkými množinami dat
- mohou být vytvořeny tak, aby vytvářely faktury, objednávky, poštovní etikety a jiné výstupy, které mohou být potřebné pro efektivní vedení podniku

Sestavy jsou navrženy tak, aby mohly data seskupovat, představit každé seskupení samostatně a provádět nad nimi výpočty.[16]



Jelikož databáze obsahuje zatím pouze data, která jsou určena k archivaci, není potřeba vytvářet sestavy s nějakými speciálními funkcemi. Bylo vytvořeno pouze několik jednoduchých sestav na základě dotazů, aby se zřehlednily výstupní informace. Tyto sestavy tedy přehledně zobrazují výsledky dotazů a jsou připravené pro případný export do jiného formátu nebo pro tisk. Všechny sestavy byly vytvořeny pomocí nástroje Sestava na kartě Vytvořit (viz obrázek 17). V příloze C jsou uvedeny dvě ukázky vytvořených sestav.



**Obrázek 17:** Vytvoření nové sestavy

*Zdroj: vlastní zpracování*

## 6. Zhodnocení

Navrhnutý relační model dat byl implementován do softwarové aplikace Microsoft Access 2007. V programu byla vytvořena nová databáze obsahující deset tabulek navzájem propojených relacemi. Do takto připravených tabulek byla podle přání majitele nahrána data týkající se zakázek z let 2009-2011. Za spoluúčasti majitele byly dále nad daty prováděny různé operace, jednak s účelem prověření konzistentnosti vytvořené databáze a zároveň naučení majitele základnímu zacházení s programem.

Jelikož vytvořená databáze zatím obsahuje pouze starší data určená k archivaci, byly vytvořeny jen základní vyhledávací dotazy a sestavy. Výsledky dotazů jsou pak přehledy za jednotlivé roky. Pro majitele bylo stěžejní získávat informace o počtech zakázek, příjmech za zakázky, nákladech a informace o pracovnících. Všechny tyto informace bylo možné na základě různých dotazů z vytvořené databáze získat, jediný problém se objevil u celkových nákladů, které nemohly být zjištěny na základě absence potřebných dat. Z výsledků byly ještě vytvořeny jednoduché sestavy pro přehlednější zobrazení.

Majitel firmy byl velice spokojený s uživatelským prostředím vybraného programu a velmi rychle pochopil základní práci v něm. Rozhodl se pro vkládání a případnou úpravu dat pomocí jednoduchých formulářů. Přehledy, které byly z databáze vytvořeny, obsahují informace, které si představoval, a poskytly mu jednotný přehled o stavu firmy za uplynulé roky.

Na základě těchto zjištění by se dalo konstatovat, že vytvořená databáze je zcela konzistentní a splňuje předem stanovené požadavky a očekávání majitele.

Pro zlepšení práce s databází do budoucna by bylo dobré vytvoření jednotného systému ukládání dat a zodpovědnému přístupu majitele k této činnosti. Například by bylo vhodné vytvoření speciálního formuláře na tvorbu nabídek tak, aby byl majitel schopný konkrétní nabídku vytvořit a následně uložit přímo v programu Access, čímž by odpadlo vytváření dalších souborů a následná práce s jejich přeukládáním do databáze. Dále by bylo dobré sjednotit systém objednávání potřebného materiálu a po skončení každé zakázky zanechat konkrétní údaje o těchto nákladech, aby mohl být majitel schopný na konci roku porovnávat výši příjmů s celkovými náklady.

## Závěr

Cílem této práce bylo vytvoření databáze pro konkrétní firmu, které zatím žádným způsobem data elektronicky nezpracovávala. Veškeré potřebné informace o zakázkách byly zaznamenávány pomocí papírové dokumentace. Po dohodě s majitelem firmy byla vytvořena klasická relační databáze na ukládání těchto dat.

V první části práce je uvedena teorie týkající se problematiky databází. Samotné tvorbě databáze v nějakém softwarovém nástroji musí předcházet kompletní model dané firemní realie a jeho následná transformace na relační model dat. Tento model pak musí být ještě normalizován, aby se zvýšila jeho spolehlivost a zabránilo se vzniku duplicitních dat. I teorie týkající se tohoto postupu je uvedena v první části práce.

Následuje analýza firmy a stávajícího stavu ukládání dat. Popsána byla firma jako taková, a jakým způsobem pak vznikají jednotlivé dokumenty, které obsahují nejdůležitější informace o zakázkách. V této části také byly stanoveny požadavky, které by databáze z hlediska majitele firmy měla splňovat.

Další část práce je věnována tvorbě konceptuálního modelu. Jsou zde definované jednotlivé entity a jejich atributy a určené vztahy, které mezi těmito entitami vznikají. Výsledkem této části je E-R diagram modelované realie. Následuje kapitola popisující transformaci vytvořeného modelu na relační model dat, při které z definovaných entit vzniknou relace, transformovány jsou i vztahy mezi původními entitami. Vzniklé relace jsou dále podle teoretických pravidel normalizovány pomocí tří základních normálních forem.

Na konci práce je vytvořený relační model dat implementován do konkrétního softwarového nástroje. Jako vhodný program byl z několika důvodů zvolen Microsoft Access. Při implementaci byly v programu vytvořeny příslušné tabulky podle relací a do nich následně uložena potřebná data. Nad daty bylo provedeno několik operací, které potvrdily konzistentnost vytvořené databáze.

Majitel firmy byl poučen tak, aby byl schopen spravovat vytvořenou databázi zcela sám a je s výsledkem práce spokojen. Na základě těchto faktů může být konstatováno, že cíl této práce byl splněn.

## Použitá literatura

- [1] CONOLLY, Thomas, BEGG, Carolyn a HOLOWCZAK Richard. *Mistrovství-databáze: Profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, a.s., 2009, 584 s. ISBN 978-80-251-2328-7.
- [2] *Databázový model* [online]. 2006 [cit. 2011-11-25]. Databázový model. Dostupné z WWW: <[http://vydavatelstvi.vscht.cz/knihy/uid\\_es005/hesla/databAzovY\\_model.html](http://vydavatelstvi.vscht.cz/knihy/uid_es005/hesla/databAzovY_model.html)>.
- [3] *ELOV Elektro Josef Vaníček* [online]. 2010 [cit. 2011-11-25]. Elov elektro. Dostupné z WWW: <<http://www.elov.cz/>>.
- [4] HERNANDEZ, Michael J. *Návrh databází*. Praha: Grada Publishing, a.s., 2006, 408 s. ISBN 80-247-0900-7.
- [5] KALUŽA, Radovan. *Konceptuální datový model* [online]. 7. 9. 2004 [cit. 2011-11-25]. Konceptuální datový model. Dostupné z WWW: <<http://owebu.bloger.cz/Databaze/Konceptualni-datovy-model?clanek=298>>.
- [6] *Konceptuální model databáze* [online]. 2006 [cit. 2012-04-21]. Konceptuální model databáze. Dostupné z WWW: <[http://vydavatelstvi.vscht.cz/knihy/uid\\_es005/hesla/konceptuAlni\\_model\\_databAze.html](http://vydavatelstvi.vscht.cz/knihy/uid_es005/hesla/konceptuAlni_model_databAze.html)>.
- [7] MATIAŠKO, Karol. *Databázové systémy*. Žilina: Žilinská univerzita, 2002, 367 s. ISBN 80-7100-968-7.
- [8] *Office.com* [online]. © 2012 [cit. 2012-04-21]. Příručka k relacím mezi tabulkami. Dostupné z WWW: <<http://office.microsoft.com/cs-cz/access-help/prirucka-k-relacim-mezi-tabulkami-HA010120534.aspx?CTT=1>>
- [9] *Office.com* [online]. © 2012 [cit. 2012-04-21]. Úvod do dotazů. Dostupné z WWW: <<http://office.microsoft.com/cs-cz/access-help/uvod-do-dotazu-HA010209892.aspx>>
- [10] *Office.com* [online]. © 2012 [cit. 2012-04-21]. Úvod k datovým typům a vlastnostem pole. Dostupné z WWW: <<http://office.microsoft.com/cs-cz/access-help/uvod-k-datovym-typum-a-vlastnostem-pole-HA010233292.aspx>>

- [11] POKORNÝ, Jaroslav; HALAŠKA, Ivan. *Databázové systémy*. Praha : Vydavatelství ČVUT, 2003. 148 s. ISBN 80-01-02789-9.
- [12] RIESSLER, Petr. *Databázové systémy*. Zlín : Vysoké učení technické v Brně, 2000. 102 s. ISBN 80-214-1778-1.
- [13] RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000, 280 s. ISBN 80-7226-360-9.
- [14] ŠEDA, Miloš. *Databázové systémy* [online]. Databázové systémy. Brno : VUT FSI v Brně, 2002 [cit. 2011-11-25]. Dostupné z WWW: <[http://www.uai.fme.vutbr.cz/~mseda/DBS02\\_BS.pdf](http://www.uai.fme.vutbr.cz/~mseda/DBS02_BS.pdf)>.
- [15] ŠIMONOVÁ, Stanislava; PANUŠ, Jan. *Databázové systémy I : pro kombinovanou formu studia*. Pardubice : Univerzita Pardubice, 2007. 106 s. ISBN 978-80-7194-988-6.
- [16] VIESCAS, John. *Mistrovství v Microsoft Access 2000: Kompletní průvodce efektivního uživatele i tvůrce databází*. 1. vyd. Praha: Computer Press, 2000, 819 s. ISBN 80-722-6274-2.
- [17] *www.manualy.net* [online]. 2. 8. 2007 [cit. 2012-04-21]. Teorie relačních databází: Normalizace. Dostupné z WWW: <<http://www.manualy.net/article.php?articleID=13>>

## Příloha A – Formulář pro vkládání dat o zakázkách

Zakázka	
<h1>Zakázka</h1>	
Číslo:	<input type="text" value="03/2009"/>
Druh práce:	<input type="text" value="SDK"/>
Název akce:	<input type="text" value="Hotel Galant"/>
Místo:	<input type="text" value="Mikulov"/>
Počet pracovníků:	<input type="text" value="5"/>
Předávací protokol:	<input type="text" value="03"/>
Datum předání:	<input type="text" value="30.4.2009"/>
IČO odběratele:	<input type="text" value="25551094"/>
Číslo smlouvy:	<input type="text" value="001/2009"/>

Záznam: 3 z 48 Bez filtru Vyhledávání



## Příloha C – Ukázky sestav

### Přehled po termínu

## Přehled zakázek předaných po termínu 2009

Číslo	Název akce	Termín dokončení	Datum předání	Název firmy
02/2009	Sodovkárna	30.12.2008	8.1.2009	D&S spol. s.r.o.
04/2009	Firemní kuchyně	9.1.2009	15.1.2009	TOKOZ a.s.
19/2009	Osvětlení ordinace	19.6.2009	22.6.2009	MVDr. Zuzana Mravíková
25/2009	KD Přibyslav	7.8.2009	10.8.2009	LEVEL Hlinsko, s.r.o.
28/2009	Veřejné osvětlení	31.8.2009	6.10.2009	CITELUM, a.s.
31b/2009	Hala PKS Mont	31.8.2009	1.9.2009	PKS INPOS a.s.
32/2009	Hala COOPER	6.8.2009	7.8.2009	PKS INPOS a.s.
35/2009	Hotel Galant	18.9.2009	21.9.2009	DOBRÁ ADRESA, s.r.o.
36/2009	ZOK	18.9.2009	22.9.2009	PKS INPOS a.s.
38/2009	Viladomy Klafar II	7.12.2009	10.12.2009	MEGALITY s.r.o.
41/2009	Hotel Galant	30.11.2009	2.12.2009	DOBRÁ ADRESA, s.r.o.

11

Stránka 1 z 1

### Přehled po splatnosti

## Přehled plateb přijatých po datu splatnosti 2009

Název firmy	Název akce	Datum splatnosti	Skutečná splatnost	Fakturovaná částka
ZOK-systém s.r.o.	Peal Liberec	18.3.2009	20.3.2009	9 480,50 Kč
Milan Mittag	ÚSP Křižanov	30.6.209	26.6.2009	81 973,50 Kč
Milan Mittag	ÚSP Křižanov	7.7.2009	8.7.2009	80 153,00 Kč
PKS INPOS a.s.	Sociální zařízení	9.7.2009	12.7.2010	2 975,00 Kč
MŠ Nové Město na Moravě	Obklad potrubí areálu školky	14.7.2009	16.7.2009	1 844,50 Kč
LEVEL Hlinsko, s.r.o.	KD Přibyslav	28.8.2009	31.8.2009	11 017,00 Kč
CITELUM, a.s.	Veřejné osvětlení	28.8.2009	31.8.2009	238 000,00 Kč
celkem záznamů				fakturovaná částka celkem

7

425 443,50 Kč

Stránka 1 z 1